Detection of Lines Using Hierarchical Region Based Representation

Alexandre Gonçalves Silva[1,2], Roberto de Alencar Lotufo[1]

[1]DCA - FEEC–Faculdade de Engenharia Elétrica e Computação - UNICAMP
P.O.Box 6101, 13083-970, Campinas, SP - Brazil
[2]DCC - CCT–Centro de Ciências Tecnológicas - UDESC
P.O.Box 631, 89223-100, Joinville, SC - Brazil
{alexgs,lotufo}@dca.fee.unicamp.br

Abstract. This paper suggests an hierarchical region based representation to implement detection of shapes in gray-scale images. The Max-Tree structure has demonstrated to be useful for this purpose, offering a semantic objects vision of the image, therefore, reducing the number of elements to process in relation to the pixel based representation. In this way, a particular thresholding information, associated with a specific shape analysis, can be used to determine the segmentation or filtering of regions of interest with lesser computational effort. For to demonstrate this technique, a generic application of detection of lines is proposed using the distance transform approach on each connected component from the depth searching in the mentioned tree.

1 Introduction

In most image processing based applications, an image is usually viewed as a set of pixels placed on a retangular grid. The developed works, in general, are based on this point of view. One of the difficulties when working directly with a pixel based representation is the large number of these elements in an image and mainly in a video sequence. But it is possible to understand an image as a set of regions. This has been done, for instance, in development of the MPEG-4 [3] or MPEG-7 [17, 14] standards. Both interpret the image or video as a set of audiovisual objects. Thus new representations of the image (not pixel based) with the implicit notion of region should be developed.

Some structures as adjacency graphs or hierarchical graphs has demonstrated to be useful for image processing based on region. Among the considered models that allow the extraction of semantic properties [14] of an image, we can cite: Region adjacency graph [5], Components Tree [12], Binary Partition Tree [16], Tree of Critical Lakes [15, 4] and Max-Tree (Min-Tree) [8].

Watershed-based [2] applications are sufficiently common. We can generate the multiscale lines of watershed from the cited tree of critical lakes. In other words, we determine objects in some scales (partitions) from coarsest to finest. Another tool very used and sufficiently powerful is the simple or adaptive thresholding that can directly be obtained by the algorithm of Max-Tree construction. This structure will be studied in this work aiming the elaboration of new algorithms for segmentation and filtering of images in gray-scales. Hough transform is frequently used for line extraction (segmentation) [7, 13]. There are works also using wavelet [9] and space-scale [1]. Our results are important task in the identifying of lines in images, such as roads and markings in digitized GIS map images and remotely sensed satellite images, among others examples.

The Section 2 shows preliminary definitions of image processing, besides introducing the Max-Tree construction. The Section 3 illustrates the proposed algorithm for detection of lines. In Section 4, we have some experimental results. Finally, conclusions are commented in Section 5.

2 Preliminary Definitions

Let $E \subset \mathbf{Z} \times \mathbf{Z}$ be a rectangular finite subset of points. Let $K = [0, k]$ be a totally ordered set. Denote by $Fun[E, K]$ the set of all functions $f : E \rightarrow K$. An *image* is traditionally one of these functions. Particularly, if $K = [0, 1]$, f is a binary image.

Let $\mathcal{N}_E(x)$ be the set containing the *neighbourhood* [11] of x, $x \in E$. We define a *path* from x to y, where $x, y \in E$, as a sequence $C = (p_0, p_1, ..., p_n)$ from E, where $p_0 = x$, $p_n = y$ and $\forall i \in [0, n-1]$, $p_i \in \mathcal{N}_E(p_{i+1})$.

A *connected component* of E is a subset $CC \subset E$ such that, $\forall x, y \in CC$, there is a path C entirely inside CC.

A *flat zone* of f is a connected component $CC \subset E$, such that $f(x) = f(y)$, $\forall x, y \in CC$.

A *regional maximum* is a flat zone Z such that $f(z) > f(n)$, $z \in Z$, $n \in N, N \in \mathcal{F}_Z$, where $\mathcal{F}_Z$ is a set of all flat zones adjacent to Z [6].

A *partition* $\mathcal{P} = \{R_1, R_2, \cdots, R_n\}$ of E is a set of regions (connected components) R_i, $i = [1, n]$, where $\bigcup R_i = E$, $R_i \cap R_j = \emptyset$, $\forall i, j \in [1, n]$, $i \neq j$.

A *partition hierarchy* or *nestled partitions sequence* $\mathcal{H} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$ is a set of partitions $\mathcal{P}_i$, $i = [1, n]$ if, for $\forall i, j \in [1, n]$, $i < j$, $\mathcal{P}_i \supset \mathcal{P}_j$.

Dilation and *erosion* are morphological operators [10] and are denoted, respectively, by δ_B and ε_B throughout this text.

Definition 1 *Let $f \in Fun[E, K]$, $K = [0, k]$, and $x \in E$, the negative operator is defined by*

$$\nu(f) = k - f(x) \ .$$

Notice that, when $k = 1$ (binary images), $\nu(f)$ is the set-complement of f.

Definition 2 *Let $f \in Fun[E, K]$, $K = [0, 1]$, $x \in E$, and B_c a structuring element, the distance transform consists in the determination of the lesser distance from x to complementary set of f, or $\nu(f)$, and is defined by*

$$DT_{B_c}(f) = \sum_{i}^{\infty} \varepsilon_{(i \cdot B_c)}(f) \ .$$

When the erosion generates a null image, the calculation of $DT_{B_c}(f)$ will be finished, and it will not be more necessary to increase the value i.

Definition 3 *Let $f, m \in Fun[E, K]$, $K = [0, k]$, and B_c a structuring element, the reconstruction is given by*

$$\begin{aligned} \gamma_{B_c,m}(f) &= \underbrace{\delta_{B_c,f}\left(\delta_{B_c,f}\left(\cdots \delta_{B_c,f}\left(f \wedge m\right)\right)\right)}_{\infty} , \\ \delta_{B_c,f}(\cdot) &= \delta_{B_c}(\cdot) \wedge f \ . \end{aligned}$$

The created image is given by an infinite number of recursive iterations (until stability) of the dilation of m by B_c conditioned to f. In other words, $\gamma_{B_c,m}(f)$ is the reconstruction of f from the marker m.

Definition 4 *Let $f \in Fun[E, K]$, $K = [0, 1]$, and B_c a structuring element, the labeling is given by*

$$\Lambda_{B_c}(f)(x) = \begin{cases} \min\{y_1 + Hy_2 | y \in \gamma_{B_c,\{x\}}(f)\} & ^{(1)} \\ 0 & ^{(2)} \end{cases}$$

$^{(1)}$ *if* $\gamma_{B_c,\{x\}}(f) \neq \emptyset$; $^{(2)}$ *otherwise. Where H is the number of image rows, y_1 and y_2 are the row and column coordinates respectively.*

Definition 5 *The morphological gradient, $\forall f \in Fun[E, K]$, and for every structuring element B_{dil} (of dilation) and B_{ero} (of erosion), is given by*

$$\Psi_{B_{dil},B_{ero}}(f) = \delta_{B_{dil}}(f) - \varepsilon_{B_{ero}}(f) \ .$$

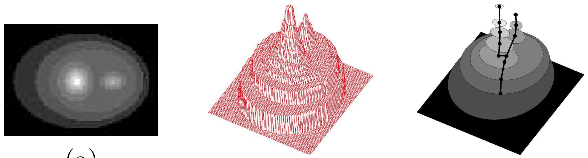
This operator is frequently used to enhance contours or non homogeneous regions.

The intention of this work is to redefine an image through a hierarchical structure where each represented connected component is a classic binary image.

2.1 Max-Tree Representation

Figure 1a shows a created synthetic image to illustrate the Max-Tree construction process. A topographical surface representation of this image is shown in Figure 1b. The Max-Tree is like a kind of skeleton of this surface, as it can be observed in Figure 1c, where each node of the tree corresponds to a connected component in a determined level of the image. The thresholding at level 0 generates all the pixels equals to 1 (unique component represented by root of the Max-Tree). When this value increases (until 255 for unsigned *byte* type images), new ramifications can appear in the tree. The levels will correspond to the regional maximum regions. Algorithm 1 implements this structure construction. The inputs are: gray-level image f, and structuring element SE (neighbourhood information). As outputs we have: Max-Tree data structure *graph*, and the flooding sequence SEQ for region retrieval. $STATUS$ is a structure that registers if a pixel already was visited. $FIFO$ represents an hierarchical queue with operations of removal ($FIFO\_FIRST$) and insertion ($FIFO\_ADD$) with a level-set of the image associated. The auxiliary vectors *node_at_level* and *number_nodes* indicate, respectively, if there are nodes and the number of nodes in each level of the image. $FIFO$ consists of a dictionary structure whose key is a level-set of the image pointing to the position and gray value of the pixel $(x, y, f(x, y))$. Finally, the *graph* structure also is a dictionary, where the key is a node $(level, label)$ of the Max-Tree pointing to its children. $APPEND$ is the function of insertion of a node in the *graph* when the key already exists and points to a list of children. Otherwise, the insertion is only an attribution. The recursive calling is done at line 18.

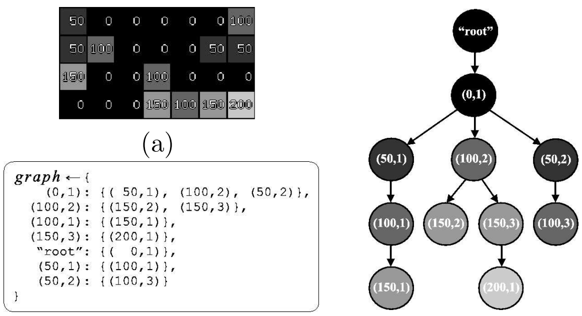
An numeric example is given in Figure 2a to enrich the visualization of this process. After the ending of the algorithm execution, we have the data structure shown in Figure 2b. It is constructed of several lists, each one referenced by a key. Each element is indicated by its level in the gray-level image, and its labeled connected component in this level (this retrieving is possible through of the output SEQ from the algorithm). Observe that the thresholding at a certain



(a) (b) (c)

Figure 1: (a) Original image (b) Topographical surface (c) Max-Tree representation

level can generate several connected components and this produces ramifications in the tree, or hierarchical partitions of the Max-Tree.



(a) (b) (c)

Figure 2: (a) Matrix of the original image (b) Data structure generated by the algorithm (c) Illustrative representation of the Max-Tree

3 The Proposed Detection of Lines

In this section, we describe each step of our proposed technique for detection of lines in gray-scale images from the Max-Tree information. The idea consists of a searching in depth on *graph* structure and, for each visited node (each connected component or region), a shape analysis is performed. We can consider that all the pixels of a line are distant of 1 unit in relation to object boundary (for lines, is convenient to use cross structuring element). An approach by distance transform can be introduced in this way: $\max(DT_{SE}(f)) = 1$. Case the line thickness is important then: $d_{min} \leq \max(DT_{SE}(f)) \leq d_{max}$. Therefore, we will search lines with thickness between d_{min} and d_{max}, or the maximum value of the distance transform must be in this interval of tolerance. The inputs of the algorithm are: original image f, the Max-Tree *graph*, the flooding sequence SEQ, and the structuring element SE (cross). Algorithm 2 describes this purpose in details, where CC_{node} is the retrieved connected component

Algorithm 1: Max-Tree construction algorithm.

INPUT: f, SE
OUTPUT: $graph$, SEQ

INITIALIZATION:
$graph \leftarrow \emptyset$; $S \leftarrow 1$
$number\_nodes \leftarrow \bigcirc_{256}$ *//zeros vector*
$node\_at\_level \leftarrow \mathcal{B}_{256}$ *//booleans vector*
$STATUS \leftarrow SEQ \leftarrow \bigcirc_{shape(f)}$
$FIFO \leftarrow 0 : \{0, 0, f[0,0]\}$

FLOOD(h)
while $h \neq \emptyset$
  $p \leftarrow FIFO\_FIRST(h)$
  $SEQ[p] \leftarrow S$
  $S \leftarrow S + 1$
  $STATUS[p] \leftarrow number\_nodes[h] + 1$
  for each $q \in \mathcal{N}_{SE}(p)$
    if $STATUS[q] =$ "not analysed"
      $m \leftarrow f[q]$
      $FIFO\_ADD(m : \{q[0], q[1], m\})$
      $STATUS[q] \leftarrow$ "in the queue"
      $node\_at\_level[m] \leftarrow TRUE$
      while $m > h$
        $m \leftarrow$ FLOOD(m)
$m \leftarrow h - 1$
while $m \geq 0$ **and** (**not** $node\_at\_level[m]$)
  $m \leftarrow m - 1$
if $m \geq 0$
  $i \leftarrow number\_nodes[h] + 1$
  $j \leftarrow number\_nodes[m] + 1$
  $UpCC_m^j \leftarrow (m, j)$
  $UpCC_h^i \leftarrow (h, i)$
  if $UpCC_m^j \in graph$
    $APPEND(graph[UpCC_m^j], UpCC_h^i)$
  else
    $graph[UpCC_m^j] \leftarrow UpCC_h^i$
else
  $UpCC_h^1 \leftarrow (\min(f), 1)$
  $graph[$"root"$] \leftarrow UpCC_h^1$
$node\_at\_level[h] \leftarrow FALSE$
$number\_nodes[h] \leftarrow number\_nodes[h] + 1$
return m

given by Algorithm 3 from a visited tree *node*. It is a model to implementation of detection of others shapes. As outputs we have: two images, g_h with the associated thresholdings to each detected object and, g_{area} with associated areas. Both images have the same connected component set with the found lines but, in addition to segmentation of lines, we can classify them in according with the criterion of level-set or area. For instance, we can filter all the detected lines with area greater than a predetermined value k: $g_{area} \geq k$.

4 Experimental Results

In this section, some examples are illustrated to demonstrate particularitities of our technique. Figure 3 shows four input images of lines with illumination gradient and distinct backgrounds, and its respective outputs

Algorithm 2: Algorithm for detection of lines.

INPUT: f, $graph$, SEQ, SE, d_{min}, d_{max}
OUTPUT: g_h, g_{area}

INITIALIZATION:
$g_h \leftarrow g_{area} \leftarrow \bigcirc_{shape(f)}$
$LIFO \leftarrow graph[graph[\text{“root”}]]$

LINE_DEPTH_SEARCH
while $LIFO \neq \emptyset$
    $node \leftarrow LIFO\_POP()$
    $LIFO\_PUSH(node)$
    $line \leftarrow d_{min} \leq \max\left(DT_{SE}\left(CC_{node}\right)\right) \leq d_{max}$
    if $line$
        $g_h \leftarrow g_h \cdot \nu(line) + node[0] \cdot line$
        $g_{area} \leftarrow g_{area} \cdot \nu(line) + area(line) \cdot line$
    else if $node \in graph$
        if $LIFO$
            $LIFO\_PUSH(graph[node])$
        else
            $LIFO \leftarrow graph[node]$
return g_h, g_{area}

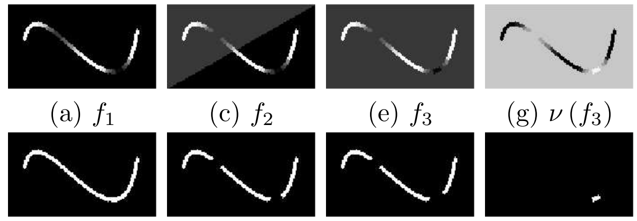
Algorithm 3: Algorithm for connected component retrieval from a node of the graph.

INPUT: f, SEQ, SE, $node$
OUTPUT: CC_{node}

REGION_RETRIEVAL
$blobs \leftarrow \gamma_{(SE, f=node[0])}(f \geq node[0])$
$label \leftarrow \Lambda_{SE}(blobs)$
for $i \leftarrow 1$ **to** $\max(label)$
    $b_i \leftarrow label = i$
    $m_i \leftarrow \min(SEQ \cdot blobs \cdot b_i)$
$n \leftarrow sort(m)$
$CC_{node} \leftarrow \{b_i | m_i = n_{node[1]}\}$
return CC_{node}

after the algorithm execution. Figure 3a shows the line with all the pixels greater than 0 and a black background, resulting in the complete detection of the line conforming Figure 3b. Figure 3c contains half of the background gray-level coincident with parts of the line, resulting in the two breaks of the line of Figure 3d (the second gap can be suppressed from Algorithm 4 - see also Figure 4). In Figure 3e, all the background assumes the same value greater than 0 and this results in the increase of the second gap as shown in Figure 3f. Finally, in Figure 3g we have the negation of the last input image and the respective output in Figure 3h. From the last example, we can observe that is interesting to process the original image and its negation for a more accurate detection by the union of both outputs (we also can use a Min-Tree structure [8] for this).

Algorithm 4 purposes a refinement of the detection of lines, when it is visually possible to find a line, but the analysis made previously does not allow to affirm that a processed connected component is a line. Suppose we have the object of Figure 4a (generated by thresholding on Figure 3c). Its distance transform corresponds to Figure 4b. Observe the maximum value is high (70) and is out of the interval for a line. When implementing this algorithm, we eliminate the regions where the distance transform is greater than pre-defined limit d_{max} of the line. Figure 4c shows this result.



(a) f_1 (c) f_2 (e) f_3 (g) $\nu(f_3)$
(b) g_1 (d) g_2 (f) g_3 (h) g_4

Figure 3: (a),(c),(e),(g) Original images; (b),(d),(f),(h) Respective results of the segmentation

Algorithm 4: Algorithm to improve the detection of lines in connected component whose distance transform maximum is greater than d_{max} limit.

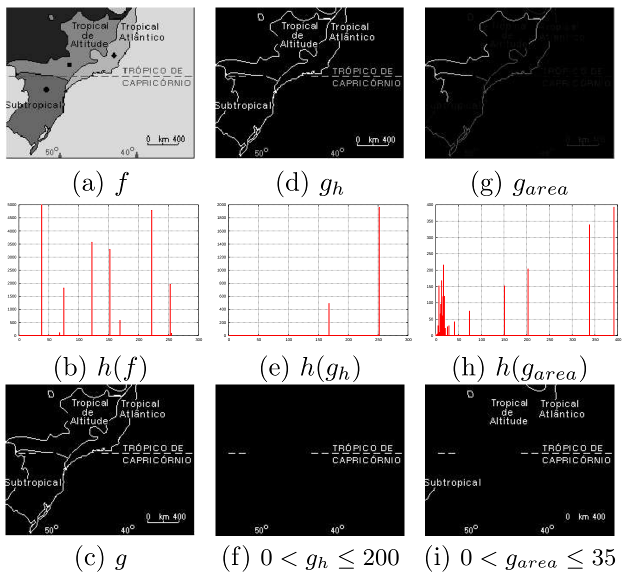
INPUT: SE, CC_{node}, d_{max}
OUTPUT: $lines$

REFINEMENT
$a_1 \leftarrow \delta_{B_{(d_{max} \cdot box)}}\left(DT_{SE}\left(CC_{node}\right) > d_{max}\right)$
$a_2 \leftarrow CC_{node} - a_1$
$a_3 \leftarrow (a_2 > 0) \cdot a_2$
$lines \leftarrow CC_{node} \cap \delta_{B_{cross}}(a_3)$
return $lines$

(a) $f_5 = f_2 > 0$ (b) $DT_{SE}(f_5)$ (c) g_5

Figure 4: (a) Thesholding of the f_2 image at level 1 (b) Distance image of the f_5, where the maximum value is 70 (c) Line retrieval by the proposed refinement

Figure 5 shows the line segmentation of portion of a Brazilian climatic map, using $d_{min} = 1$ and $d_{max} = 2$. Figure 5a,b,c are: the original image, its histogram, and the detection resulting (thresholding at level 1 in g_h or g_{area} output). Observe circles, rectangles, triangles and other shapes are not considered. Figure 5d,e shows: g_h with associated level-sets in each detected line, and its histogram. We can perceive this histogram $h(g_h)$ has only two components and to take a thresholding decision is facilitated in this format in relation to original image. For instance, we take the level 200 and generated Figure 5f image. Figure 5g,h shows: g_{area} with associated areas in each detected line, and its histogram. Figure 5i shows a filtering by area of value 35 based on generated g_{area} and its histogram $h(g_{area})$.



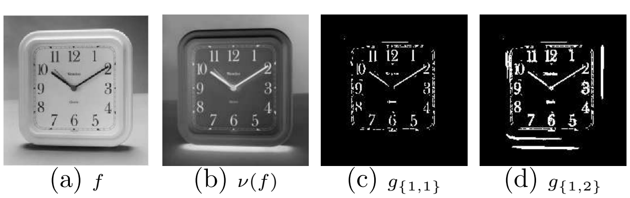
(a) f (d) g_h (g) g_{area}

(b) $h(f)$ (e) $h(g_h)$ (h) $h(g_{area})$

(c) g (f) $0 < g_h \leq 200$ (i) $0 < g_{area} \leq 35$

Figure 5: (a) Original image f (b) Histogram of f (c) Result of the segmentation (d) Output g_h (e) Histogram of g_h (f) Threshold 200 of g_h (g) Output g_{area} (h) Histogram of g_{area} (i) Threshold 35 of g_{area}

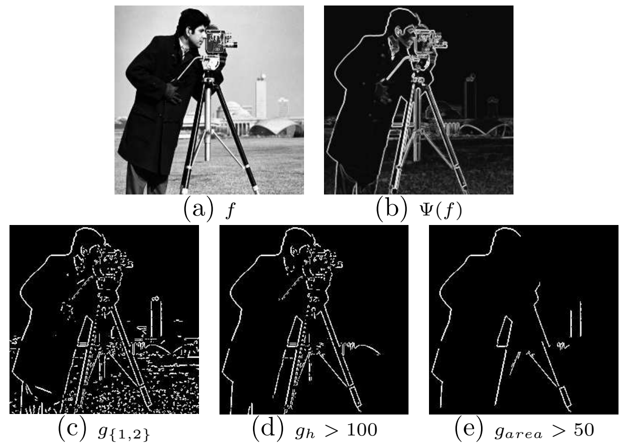
Our next example is the purposed algorithm application of line detection on a photographic image of a clock. Figure 6a shows the original image. We aim to segment dark lines and, therefore, we take the negation of the clock image gererating Figure 6b. For $d_{min} = 1$ and $d_{max} = 1$, $g_h > 0$ or $g_{area} > 0$ corresponds to Figure 6c image. For $d_{min} = 1$ and $d_{max} = 2$, we have the segmentation of Figure 6d image. In the last case, thicker lines was found.



(a) f (b) $\nu(f)$ (c) $g_{\{1,1\}}$ (d) $g_{\{1,2\}}$

Figure 6: (a) Original image f (b) Negation of f (c) Binary ouput for $d_{min} = 1$ and $d_{max} = 1$ (d) Binary ouput for $d_{min} = 1$ and $d_{max} = 2$

Finally, the last example consists in detection of lines in another photographic image of a cameraman. The original image is in Figure 7a. Its gradient is used to enhancement the contours and is shown in Figure 7b. The algorithm is applied resulting in Figure 7c. Two thresholdings are tested to illustrate some outputs in Figure 7d and Figure 7e.



(a) f (b) $\Psi(f)$

(c) $g_{\{1,2\}}$ (d) $g_h > 100$ (e) $g_{area} > 50$

Figure 7: (a) Original image f (b) Gradient of f (c) Binary ouput for $d_{min} = 1$ and $d_{max} = 2$ (d) Threshold 100 of g_h (e) Threshold 50 of g_{area}

5 Conclusions

The presented Max-Tree algorithm is fast with runtime of $k \times n$ on gray-scale images of n pixels. k comparisons, where k is the number of neighbours of a pixel (line 11 of Algorithm 1), are performed for each pixel extracted from hierarchical queue. The proposed algorithm implements linear time operations, such as the distance transform, on each node of the subtree of the Max-Tree. The region retrieving method (Algorithm 3) still can be improved. The classical Hough transform is very used in detection of lines and does $m \times p$ computations, being m the number of points in a binary image (therefore, a pre-processing is necessary) and p the number of subdivisions of the space of parameters. Several variants were introduced to improve perfomance in classical Hough transform, which has high complexity in the practice.

During the line detection processing, whenever a line is found the descendent nodes do not need be visited. Figure 8 demonstrates this feature. A performance gain is obtained by this issue. Instead of line analysis we could easily take a circle, an ellipse, a rectangle or any other simple geometric figure. Observe these examples were not detected because are not included in the distance transform approach adopted here. More complex objects can also be detected by statistical analysis (classification) or other specific mechanisms (contour derivation, convex hull, neural networks, among others).

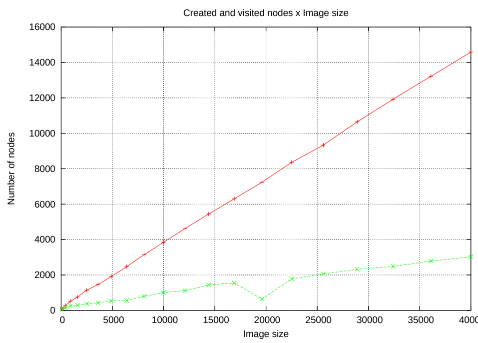


Figure 8: Number of created nodes ("+" markers) by Max-Tree and number of visited nodes ("x" markers) by proposed algorithm for several random images with increasing size

The generated image g_h has simplified histogram and a thresholding decision is facilitated, aiming to find lines with a certain gray-level. The output g_{area} is useful in filtering of lines by area. If $d_{min} = 1$ and $d_{max} = 1$ then a classification by line length can be performed. The line thickness is not normally considered and only straight lines are usually extracted in works using Hough transform. This paper offers an image semantic vision allowing to detect any line conform to a defined thickness. Moreover, the proposed algorithm can be expanded to segment other shapes or objects.

For each visited tree node, we must know the connected component correspondent. Further work is required to avoid any redundant processing of connected component retrieval as, for instance, the labeling or distance transform in a determined image thresholding. Information as these would be registered only once and consulted when necessary. The algorithm could also generate a labeled image g_{label}, characterizing each line and, from this, g_h and g_{area} could be created. Intrinsic Max-Tree characteristics will be studied for segmentation of other objects and filtering. Line detection in other hierarchical region based representation as the Tree of Critical Lakes must also be implemented for comparisons.

6 Acknowledgments

The first author is grateful to CAPES for financial support of this work.

References

[1] W. Beil. Line detection in discrete scale-space. *ICIP-I*, 94:915–918.

[2] S. Beucher and F. Meyer. *Mathematical Morphology in Image Processing*, chapter 12. The Morphological Approach to Segmentation: The Watershed Transformation, pages 433–481. Marcel Dekker, 1992.

[3] R. Chiariglione. MPEG and Multimedia Communications. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):5–18, February 1997.

[4] M. A. G de Carvalho. *Análise Hierárquica de Imagens Através da Árvore dos Lagos Críticos*. Dissertação de Doutorado, Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas, Janeiro 2004.

[5] J. Robinson E. Costanza. A Region Adjacency Tree Aproach to the Detection and Design of Fiducials. In *Vision, Video, and Graphics 2003*, pages 63–69, Bath, UK, July 2003. Eurographics Association.

[6] F. C. Flores. Segmentação de Seqüências de Imagens por Morfologia Matemática. Dissertação de Mestrado, Instituto de Matemática e Estatística - Universidade de São Paulo, Outubro 2000.

[7] Y. Furukawa and Y. Shinagawa. Accurate and robust line segment extraction by analyzing distribution around peaks in hough space. *CVIU*, 92(1):1–25, October 2003.

[8] L. O. Garrido. *Hierarchical region based processing of images and video sequences: application to filtering, segmentation and information retrieval*. PhD thesis, Department of Signal Theory and Communications - Universitat Politcnica de Catalunya, Barcelona, April 2002.

[9] C. Hagelberg and J. Helland. Thin-line detection in meteorological radar images using wavelet transforms. *Journal of Atmospheric and Oceanic Technology*, 12:633–642, 1995.

[10] H. J. A. M. Heijmans. Introduction to Connected Operators. In E. R. Dougherty and J. T. Astola, editors, *Nonlinear Filters for Image Processing*, pages 207–235. SPIE–The International Society for Optical Engineering, 1999.

[11] R. Hirata Jr. Segmentação de Imagens por Morfologia Matemática. Dissertação de Mestrado, Instituto de Matemática e Estatística - USP, março 1997.

[12] R. Mathieu J. Mattes and J. Demongeot. Tree Representation for Image Matching and Object Recognition. In *8th International Conference on Discrete Geometry for Computer Imagery*, pages 298–312, Marne-la-Vallee, France, March 1999.

[13] J. V. Kittler J. Princen, J. Illingworth. A hierarchical approach to line extraction. In *CVPR89*, pages 92–97, 1989.

[14] M. Abdel-Mottaleb, N. Dimitrova, L. Agnihotri, S. Dagtas, S. Jeannin, S. Krishnamachari, T. McGee and G. Vaithilingam. MPEG 7: A Content Description Standard Beyond Compression. In *IEEE Midwest Symposium on Circuits and System*, New Mexico, USA, August 1999.

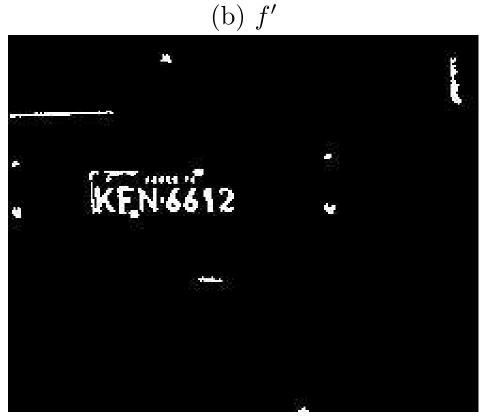
[15] M. Carvalho, R. Lotufo and M. Couprie. Spatiotemporal Segmentation of MR Image Sequence Based on Hierarchical Analysis. In *Seventh International Symposium on Signal Processing and Its Applications*, pages 677–680, Paris, France, July 2003.

[16] P. Salembier and L. Garrido. Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, 2000.

[17] T. Sikora. The MPEG-7 Visual Standard for Content Description - and Overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696–702, June 2001.

(a) f

(b) f'



(c) $g_{\{1,2\}}$

Figure 9: Application of vehicle plate detection (a) Original image (b) Reduction of contrast and negation of the original image (c) Result of the proposed algorithm applied on the second image using $d_{min} = 1$ and $d_{max} = 2$