Pattern Recognition Letters 13 (1992) 5-12
North-Holland
January 1992

Performance characterization in image analysis: thinning, a case in point

Robert M. Haralick

University of Washington, Seattle, WA 98195, USA

Received 26 November 1991

1. Introduction

Image analysis algorithms are composed of different sub-algorithms often applied in sequence. Determination of the performance of a total image analysis algorithm is possible if the performance of each of the sub-algorithm constituents is given. The problem, however, is that for most published algorithms, there is no performance characterization which has been established in the research literature. This is an awful state of affairs for the engineers whose job it is to design and build image analysis or machine vision systems.

This suggests that there has been a cultural deficiency in the image analysis/machine vision community: image analysis and processing algorithms have been published more on the merit of an experimental or theoretical demonstration suggesting that some task can be done, rather than on an engineering basis. Such a situation was tolerated because the interesting question was whether it was possible at all to accomplish an image analysis task. Performance was a secondary issue.

Now, however, a major interesting question is how to quickly design image analysis systems which work efficiently and which meet requirements. To do this requires an engineering basis which describes precisely what is the task to be done, how this task can be done, what is the error criterion, and what is the performance of the algorithm under various kinds of random degradations of the input data. To accomplish this for adaptive algorithms requires being able to do a closed loop engineering analysis. To perform a closed loop engineering analysis requires being able to first do an open loop engineering analysis.

The purpose of this discussion is to raise our sensitivity to these issues so that our field can more rapidly transfer the research technology to a factory floor technology. To initiate this dialogue, we will first expand on the meaning of performance characterization in general, then discuss the experimental protocol under which an algorithm performance can be characterized, and finally specialize the discussion to the area of thinning algorithms as a case in point.

2. Performance characterization

What does performance characterization mean for an algorithm which might be used in an image analysis or machine vision system? The algorithm is designed to accomplish a specific task. If the input data is perfect and has no noise and no random variation, the output produced by the algorithm ought also to be perfect. Otherwise, there is something wrong with the algorithm. So measuring how well an algorithm does on perfect input data is not interesting. Performance characterization has to do with establishing the correspondence of the random variations and imperfections which the algorithm produces on the output data caused by the random variations and the imperfections on the input data. This means that to do performance characterization, we must first specify a model for the ideal world in which only perfect data exist. Then we must give a random perturbation model which specifies how the imperfect perturbed data arises from the perfect data. Finally, we need a criterion function which quantitatively measures the difference between the ideal output arising from the perfect ideal input and the calculated output arising from the corresponding randomly perturbed input.

Now we are faced with an immediate problem relative to the criterion function. It is typically the case that an algorithm changes the data unit. For example, an edge-linking process changes the data from the unit of pixel to the unit of a group of pixels. An arc segmentation/extraction process applied to the groups of pixels produced by an edge linking process produces fitted curve segments. This data unit change means that the representation used for the random variation of the output data set may have to be entirely different than the representation used for the random variation of the input data set. In our edge-linking/arc extraction example, the input data might be described by the false alarm/misdetection characteristics produced by the preceding edge operation, as well as the standard deviation in the position and orientation of the correctly detected edge pixels. The random variation in the output data from the extraction process, on the other hand, must be described in terms of fitting errors (random variation in the fitted coefficients) and segmentation errors. Hence, the criterion function may change from stage to stage in the analysis process.

Consider the case for segmentation errors. The representation of the segmentation errors must be natural and suitable for the input of the next process in high-level vision which might be a model-matching process, for example. What should this representation be to make it possible to characterize the identification accuracy of the model matching as a function of the input segmentation errors and fitting errors? Questions like these, have typically not been addressed in the research literature. Until they are, analyzing the performance of an image analysis algorithm will be in the dark ages of an expensive experimental trial-and-error process. And if the performance of the different pieces of a total algorithm cannot be used to determine the performance of the total algorithm, then there cannot be an engineering design methodology for machine vision systems.

This problem is complicated by the fact that there are many instances of algorithms which compute the same sort of information but in forms which are actually non-equivalent. For example, there are arc extraction algorithms which operate directly on the original image along with an intermediate vector file obtained in a previous step and which output fitted curve segments. There are other arc extraction algorithms which operate on groups of pixels and which output arc parameters such as center, radius, and endpoints in addition to the width of the original arc.

What we need is the machine vision analog of a system's engineering methodology. This methodology can be encapsulated in a protocol which has a modeling component, an experimental component, and a data analysis component. The next section describes in greater detail these components of an image analysis engineering protocol.

3. Protocol

The modeling component of the protocol consists of a description of the world of ideal images, a description of a random perturbation model by which non-ideal images arise, and a specification of the criterion function by which the difference between the ideal output and the computed output arising from the imperfect input can be quantified. The experimental component describes the experiments performed under which the data relative to the performance characterization can be gathered. The analysis component describes what analysis must be done on the experimentally observed data to determine the performance characterization.

3.1. Image generation

This part of the protocol describes how, in accordance with the specified model, a suitably random, independent, and representative set of images from the population of ideals is to be acquired or generated to constitute the sampled set of images. This aquisition can be done by taking real images under the specified conditions or by generating synthetic images. If the population includes, for example, a range of sizes of the object of interest or if the object of interest can appear in a variety of situations, or if the object shape can have a range of variations, then the sampling mechanism must assure that a reasonable number of images are sampled with the object appearing in sizes, orientations, and shape variations throughout its permissible range. Similarly, if the object to be recognized or measured can appear in a variety of different lighting conditions which create a similar variety in shadowing, then the sampling must assure that images are acquired with the lighting and shadowing varying throughout its permissible range.

Some of the variables used in the image generation process are ones whose values will be estimated by the image analysis algorithm. We denote these variables by $z_1, \ldots, z_K$. Other of these variables are nuisance variables. Their values provide for variation. The performance characterization is averaged over their values. We denote these variables by $w_1, \ldots, w_M$. Other variables specify the state of the controlled random perturbation and noise against which the performance is to be characterized. We denote these variables by $y_1, \ldots, y_J$. The generation of the images in the population can then be described by $N = J + K + M$ variables. If these N variables having to do with kind of lighting, light position, object position, object orientation, permissable object shape variations, undesired object occlusion, environmental clutter, distortion, noise etc., have respective range sets $R_1, \ldots, R_N$, then the sampling design must assure that images are selected from the domain $R_1 \times R_2 \times \cdots \times R_N$ in a representative way. Since the number of images sampled is likely to be a relatively small fraction of the number of possibilities in $R_1 \times R_2 \times \cdots \times R_N$, the experimental design may have to make judicious use of a Latin square layout.

3.2. *Random perturbation and noise*

Specification of random perturbation and noise is not easy because the more complex the data unit, the more complex the specification of the random perturbation and noise. Each specification of randomness has two potential components. One component is a small perturbation component which affects all data units. It is often reasonable to model this by an additive Gaussian noise process on the ideal values of the data units. This can be considered to be the small variation of the ideal data values combined with observation or measurement noise. The other component is a large perturbation component which affects only a small fraction of the data units. For simple data units it is reasonable to model this by replacing its value by a value having nothing to do with its true value. Large perturbation noise on more complex data units can be modeled by fractionating the unit into pieces and giving values to most of the pieces which would follow from the values the parent data unit had and giving values to the remaining pieces which have nothing to do with the values the original data unit had.

This kind of large random perturbation affecting a small fraction of units is replacement noise. It can be considered to be due to random occlusion, linking, grouping, or segmenting errors. Algorithms which work near perfectly on small amounts of random perturbation on all data units, often fall apart with large random perturbation on a small fraction of the data units. Much of the performance characterization of a complete algorithm will be specified in terms of how much of this replacement kind of random perturbation the algorithm can tolerate and still give reasonable results. Algorithms which have good performance even with large random perturbation on a small fraction of data units can be said to be robust.

3.3. *Performance characterization*

Some of the variables used in the image generation are those whose values are to be estimated by the machine vision algorithm. Object kind, location, and orientation are prime examples. The values of such variables do not make the recognition and estimation much easier or harder, although they may have some minor effect. For example, an estimate of the surface normal of a planar object viewed at a high slant angle will tend to have higher variance than an estimate produced by the planar object viewed at a near normal angle. The performance characterization of an image analysis algorithm is not with respect to this set of variables. From the point of view of what is to be calculated, this set of variables is crucial. From the point of view of performance characterization, the values for the variables in this set as well as the values in the nuisance set are the ones over which the performance is averaged.

Another set of variables characterize the extent of random perturbations which distort the ideal input data to produce the imperfect input data. These variables represent variations which degrade the information in the image, thereby increasing the uncertainty of the estimates produced by the algorithm. Such variables may characterize object contrast, noise, extent of occlusion, complexity of background clutter, and a multitude of other factors which instead of being modeled explicitly are modeled implicitly by the inclusion of random shape perturbations applied to the set of ideal model shapes.

Finally, there may be other variables governing parameter constants that must be set in the image analysis algorithm. The values of these variables may to a large or small extent change the performance of the algorithm.

The variables governing the extent of random perturbations and the variables which are the algorithm parameter constants constitute the set of variables in terms of which the performance characterization must be measured. Suppose there are I algorithm parameters $x_1, \ldots, x_I$, which can be set, J different variables $y_1, \ldots, y_J$ governing extent of random perturbations, and K different measurements $\hat{z}_1, \ldots, \hat{z}_K$ to be made on each image. There will be a difference between the true ideal values $z_1, \ldots, z_K$ of the measured quantities and the measured values $\hat{z}_1, \ldots, \hat{z}_K$ themselves. The error criterion, $e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)$, must state how the comparison between the ideal values and the measured values will be evaluated. Its value will be a function of the I algorithm parameters and the J random perturbation parameters.

An algorithm can have two different dimensions to the error criterion. To explain these dimensions, consider algorithms which estimate some parameter such as position and orientation of an object. One dimension the error criterion can have is reliability. An estimate can be said to be reliable if the algorithm is operating on data that meets certain requirements and if the difference between the estimated quantity and the true but known value is below a user specified tolerance. An algorithm can estimate whether the results it produces are reliable by making a decision on estimated quantities which relate to input data noise variance, output data covariance, and structural stability of calculation. Output quantity covariance can be estimated by estimating the input data noise variance and propagating the error introduced by the noise variance into the calculation of the estimated quantity. Hence the algorithm itself can provide an indication of whether the estimates it produces have an uncertainty below a given value. High uncertainties would occur if the algorithm can determine that the assumptions about the environment producing the data or the assumptions required by the method are not being met by the data on which it is operating or if the random perturbation in the quantities estimated is too high to make the estimates useful.

Characterizing this dimension can be done by two means. The first is by the probability that the algorithm claims reliability as a function of algorithm parameters and parameters describing input data random perturbations. The second is by misdetection false alarm operating curves. A misdetection occurs when the algorithm indicates it has produced a reliable enough result when in fact it has not produced a reliable enough result. A false alarm occurs when the algorithm indicates that it has not produced a reliable enough result when in fact it has produced a reliable enough result. A misdetection false alarm rate operating curve results for each different noise and random perturbation specification. The curve itself can be obtained by varying the algorithm tuning constants, one of which is the threshold by which the algorithm determines whether it claims the estimate it produces is reliable or not.

The second dimension of the error criterion would be related to the difference between the true value of the quantity of interest and the estimated value. This criterion would be evaluated only for those cases where the algorithm indicates that it produces a reliable enough result.

Each estimated quantity $\hat{z}_k$ is a function of the values of the algorithm constants $x_1, \ldots, x_I$ and the random perturbation induced on the image by the values of the variables $y_1, \ldots, y_J$ and each z_k is a function only of the algorithm constants $x_1, \ldots, x_I$. The expected value E of $e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)$ is, therefore, a function of $x_1, \ldots, x_I$ and $y_1, \ldots, y_J$. Performance characterization of the estimated quantity then amounts to expressing in graph, table or analytic form $E[e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)]$ as a function of $x_1, \ldots, x_I$ and $y_1, \ldots, y_J$.

3.4. Experiments

In a complete design, the values for the algorithm constants $x_1, \ldots, x_I$ and the values governing the random perturbations $y_1, \ldots, y_J$ will be selected in a systematic and regular way. The values for $z_1, \ldots, z_K$ and the values for the nuisance variables $w_1, \ldots, w_M$ will be sampled from a uniform distribution over the range of their permissible values.

The values for $z_1, \ldots, z_K$ uniquely specify an ideal image. The values for $y_1, \ldots, y_J$ specify the extent to which random perturbations and noise are randomly introduced into the ideal image and/or object(s) in the ideal image. In this manner, each noisy trial image is generated. The values for $x_1, \ldots, x_I$ specify how to set the parameter constants required by the algorithm. The algorithm is then run over the trial image producing estimated values $\hat{z}_1, \ldots, \hat{z}_K$ for $z_1, \ldots, z_K$. Applying the error criterion then produces the values $e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)$. The data produced by each trial then consists of a record

$$x_1, \ldots, x_I, y_1, \ldots, y_J, e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K).$$

The data analysis plan describes how the set of records produced by the experimental trials will be processed or analyzed to compactly express the performance characterization. For example, an equivalence relation on the range space for $y_1, \ldots, y_J$ may be defined and an hypothesis may be specified stating that all combinations of values of $y_1, \ldots, y_J$ in the same equivalence class have the same expected error. The data analysis plan would specify the equivalence relation and give the statistical procedure by which the hypothesis could be tested. Performing such tests are important because they can reduce the number of variable combinations which have to be used to express the performance characterization. For example, the hypothesis that all other variables being equal, whenever y_{J-1}/y_J has a ratio of k, then the expected performance is identical. In this case, the performance characterization can be compactly given in terms of k and $y_1, \ldots, y_{J-2}$.

Once all equivalence tests are complete, the data analysis plan would specify the kinds of graphs or tables employed to present the experimental data. It might specify the form of a simple regression equation by which the expected error, the probability of claimed reliability, the probability of misdetection, the probability of false alarm, and the computational complexity or execution time can be expressed in terms of the independent variables $x_1, \ldots, x_I, y_1, \ldots, y_J$. As well it would specify how the coefficients of the regression equation could be calculated from the observed data.

Finally, if the image analysis algorithm must meet certain performance requirements, the data analysis plan must state how the hypothesis that the algorithm meets the specified requirement will be tested. The plan must be supported by a theoretically developed statistical analysis which shows that an experiment carried out according to the experimental design and analyzed according to the data analysis plan will produce a statistical test itself having a given accuracy. That is, since the entire population of images is only sampled, the sampling variation will introduce a random fluctuation in the test results. For some fraction of experiments carried out according to the protocol, the hypothesis to be tested will be accepted but the algorithm, in fact, if it were tried on the complete population of image variations, would not meet the specified requirements; and for some fraction of experiments carried out according to the protocol, the hypothesis to be tested will be rejected but if the algorithm were tried on the complete population of image variation, it would meet the specified requirements. The specified size of these errors of false acceptance and missed acceptance will dictate the number of images to be in the sample for the test. This relation between sample size and false acceptance rate and missed acceptance rate of the test for the hypothesis must be determined on the basis of statistical theory. One would certainly expect that the sample size would be large enough so that the uncertainty caused by the sampling would be below 20%.

For example, suppose the error rate of a quantity estimated by an image analysis algorithm is defined to be the fraction of time that the estimate is further than ε_0 from the true value. If this error rate is to be less than 1/1,000, then in order to be about 85% sure that the performance meets specification, 10,000 tests will have to be run. If the image analysis algorithm performs incorrectly 9 or fewer times, then we can assert that with 85% probability, the image analysis algorithm meets specification (Haralick, 1989).

4. Thinning

In this section we take up a case in point: thinning. There are a large number of papers which have been published in this area. Each improves on the other in some fashion, but a mathematically precise statement of what is really being calculated is not made in any of them.

To illustrate this, we will sketch out one kind of ideal shape model, one kind of a random perturbation model, and one experimental protocol which concretely describes how one kind of performance characterization can be done. The conceptual model part of the protocol describes the ideal shapes which are to be thinned. Then it describes a random perturbation model which specifies the noise and distortions these shapes undergo in observation, and finally it gives a statement of the error criterion which evaluates the difference between the computed answer and the ideal answer.

4.1. The ideal shape world

The ideal world of shapes to be thinned is the world of ribbons. An ideal ribbon is constructed from a simple bounded curvature arc which is the spine, specifying the center of the ribbon and a cross-section function giving the ribbon's width at each point of the spine. Simple means that the arc does not cross itself. In addition, all arcs have bounded curvature. At any point of the arc, the width of the ribbon is the length of the line segment defined by the intersection of the ribbon with a line perpendicular to the arc at the given arc point. The cross-section function has bounded first derivatives to keep the width from changing too fast. The width itself is also bounded from both sides, to prevent too narrow or too wide ribbons. Additionally, there is a relation between the maximum allowed curvature (minimum local radius) of the arc and the maximum allowed width of the ribbon, in order to prevent a case in which the combination of sharp curvature and large width at that point cause the ribbon to overlap itself.

A scene of ribbons is a set of ribbons which do not interfere too much with one another. Non-interference means that in any scene, the total of both the length and area of the ribbons is bounded. Furthermore ribbons either do not touch one another or if they do, they touch one another only in a constrained way: if they touch, they must cross one another and the crossing must be at not too shallow an angle.

An ideal digital image is constructed from a scene of ribbons by tesellating the scene of ribbons into pixels. Each pixel is independently given a value of 0 or 1 according to a stochastic mechanism in which the probability that the value of a pixel is 1 is the fraction of the area of the pixel occupied by the ribbon.

4.2. The observed noisy perturbed shapes

The observed digital image is obtained from the ideal digital image by changing the value of a ribbon pixel with a probability depending on the distance the ribbon pixel center is from the ribbon boundary. This probability is a given monotonically decreasing function of the distance the ribbon pixel is from the ribbon boundary. Thus, pixels well into the interior of a ribbon have little likelihood of changing value. Pixels near a ribbon boundary have larger probability of changing value.

4.3. Error criterion function

Thinning can have a variety of purposes such as the estimation of the analytic expression for the spine of the ribbon or the identification of the pixels through which the spine passes. In this discussion we take the purpose of thinning to be the identification of the pixels through which the spine passes.

We define the error criterion function to be a given convex combination of the number of pixels identified by the thinning algorithm as being on the spine but which are really not on the spine with the number of pixels identified by the thinning algorithm as not being on the spine but which really are on the spine.

4.4. Discussion

It may be argued whether the outlined world of ideal ribbons is the most appropriate one, whether the purpose of thinning is most appropriately specified as the identification of the spine pixels, whether the random perturbation model is realistic, and if the criterion function is the most useful one. For the purpose of this discussion, these arguments are irrelevant: we are not trying to promote any of these choices; only to show that these are choices which must be made.

In the thinning literature, [1-45], the ideal world of ribbons is not specified, the random perturbation model is not discussed, and the error criterion function is not given. And for this reason, what precise problem any thinning algorithm solves is not, in fact, precisely stated. For a precise statement would have the form: under the given ideal world, random perturbation model, and error criterion, the following algorithm, with smallest error, identifies the pixels on the spine.

5. Conclusion

This paper has discussed the problem of the lack of performance evaluation in image analysis algorithms. This situation is causing great difficulties to researchers who are trying to build up on existing algorithms and to engineers who are designing operational systems. To remedy the situation, we suggested the establishment of a well-defined protocol for determining the performance characterization of an algorithm. Use of this kind of protocol will make using engineering system methodology possible as well as making possible well-founded comparisons between image analysis and machine vision algorithms that perform the same tasks. We elected a rather common and basic operation, namely thinning, as a first candidate to demonstrate the feasibility of our approach. With this case, a specific protocol model has been suggested for generating the test images, for the perturbation model, and for the testing criterion. We hope that our discusstion will encourage a thorough and overdue dialogue in the field so that a complete engineering methodology for performance evaluation of image analysis algorithms can finally result.

References

[1] Arcelli, C. (1981). Pattern thinning by contour tracing. *Computer Graphics and Image Processing* 17, 130-144.

[2] Arcelli, C., L.P. Cordella, and S. Levialdi (1981). From local maxima to connected skeletons. *IEEE Trans. Pattern Anal. Machine Intell.* 3, 134-143.

[3] Arcelli, C., and G. Sanniti di Baja (1985). A width-independent fast thinning algorithm. *IEEE Trans. Pattern Anal. Machine Intell.* 7, 463-474.

[4] Arcelli, C., and G. Sanniti di Baja (1981). A thinning algorithm based on prominence detection. *Pattern Recognition* 13, 225-235.

[5] Arcelli, C., and G. Sanniti di Baja (1989). A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform. *IEEE Trans. Pattern Anal. Machine Intell.* 11, 411-414.

[6] Badie, K., and M. Shimura (1985). A structural approach to extraction of line segments in binary images. *Proceedings of the Fourth Scandinavian Conference on Image Analysis*, Trondheim, 655-662.

[7] Bel-Lan, A. and L. Montoto (1981). A thinning transform for digital images. *Signal Processing* 3, 37-47.

[8] Bertrand, G. (1984) Skeletons in derived grids. *Seventh International Conference on Pattern Recognition*, Montreal, 326-329.

[9] Bourbaks, N.G. (1989). A parallel-symmetric thinning algorithm. *Pattern Recognition* 22, 387-396.

[10] Chen, Y.-S., and W.-H. Hsu (1989). A systematic approach for designing 2-subcycle and pseudo 1-subcycle parallel thinning algorithms. *Pattern Recognition* 22, 267-282.

[11] Chin, R.T., H.-K. Wan, D.L. Stover, and R.D. Iverson (1987). A one-pass thinning algorithm and its parallel implementation. *Computer Graphics, Vision, and Image Processing* 40, 30-40.

[12] Davies, E.R., and A.P.N. Plummer (1981). Thinning algorithms: a critique and a new methodology. *Pattern Recognition* 14, 53-63.

[13] Dyer, C.R., and A. Rosenfeld (1977). Thinning algorithms for grayscale pictures. University of Maryland, Computer Science Technical Report TR610.

[14] Favre, A., and Hj. Keller (1983). Parallel syntactic thinning by recoding of binary pictures. *Computer Graphics, Vision, and Image Processing* 23, 99-112.

[15] Govindan, V.K., and A.P. Shivaprasad (1987). A pattern adaptive thinning algorithm. *Pattern Recognition* 20, 623-637.

[16] Hall, R.W. (1989). Fast parallel thinning algorithms: parallel speed and connectivity preservation. *Comm. ACM* 32, 124-131.

[17] Haralick, R.M. (1989). Performance assessment of near perfect machines. *Machine Vision and Applications* 2 (1), 1-16.

[18] Hilditch, C.J. (1969). Linear skeletons from square cupboards. In: B. Meltzer and D. Michie, Eds., *Machine Intelligence IV*. University Press, Edinburgh, 403-420.

[19] Hilditch, C. J. (1983). Comparison of thinning algorithms on a parallel processor. *Image and Vision Computing* 1, 115-132.

[20] Holt, C.M., A. Stewart, M. Clint, and R.H. Perrott (1987). An improved parallel thinning algorithm. *Comm. ACM* 30, 156-160.

[21] Jang, B.K., and R.T. Chin (1990). Analysis of thinning algorithms using mathematical morphology. *IEEE Trans. Pattern Anal. Machine Intell.* 12, 541-551.

[22] Klein, F., and O. Kübler (1987). Euclidean distance transformations and model-guided image interpretation. *Pattern Recognition Letters* 5, 19-29.

[23] Kong, T.Y., and A. Rosenfeld (1989). Digital topology: introduction and survey. *Computer Graphics, Vision, and Image Processing* 48, 357-393.

[24] Kwok, P.C.K. (1988). A thinning algorithm by contour generation. *Comm. ACM* 31, 1314-1324.

[25] Martínez-Pérez, M.P., J. Jiménez, and J.L. Navalón (1987). A thinning algorithm based on contours. *Computer Graphics, Vision, and Image Processing* 39, 186-201.

[26] Naccache, N.J., and R. Shinghal (1984). SPTA: a proposed algorithm for thinning binary patterns. *IEEE Trans. Syst. Man Cybernet.* 14, 409-418.

[27] Naccache, N.J., and R. Shinghal (1984). An investigation into the skeletonization approach of Hilditch. *Pattern Recognition* 17, 279-284.

[28] Ogawa, H., and K. Taniguchi (1982). Thinning and stroke segmentation for handwritten Chinese character recognition. *Pattern Recognition* 15, 299-308.

[29] O'Gorman, L. (1990). $k \times k$ Thinning. *Computer Vision, Graphics, and Image Processing*, 195-215.

[30] Pavlidis, T. (1982). An asynchronous thinning algorithm. *Computer Graphics and Image Processing* 20, 133-157.

[31] Pavlidis, T. (1980). A thinning algorithm for discrete binary images. *Computer Graphics and Image Processing* 13, 142-157.

[32] Rearick, T.J. (1985). Syntactical methods for improvement of the medial axis transformation. *SPIE 548: Applications of Artificial Intelligence II*, 110-115.

[33] Shanmugam, K.S., and C. Paul (1982). A fast edge thinning operator. *IEEE Trans. Syst. Man Cybernet.* 12, 567-569.

[34] Stefanelli, R., and A. Rosenfeld (1971). Some parallel thinning algorithms for digital pictures. *J. ACM* 18, 255-264.

[35] Stentiford, F.W.M., and R.G. Mortimer (1983). Some new heuristics for thinning binary handprinted characters for OCR. *IEEE Trans. Syst. Man Cybernet.* 13, 81-84.

[36] Suzuki, S., and K. Abe (1987). Binary picture thinning by an iterative parallel two-subcycle operation. *Pattern Recognition* 20, 297-307.

[37] Suzuki, S., and K. Abe (1986). Sequential thinning of binary pictures using distance transformation. *Eighth International Conference on Pattern Recognition*, 289-292.

[38] Tamura, H. (1978). A comparison of line thinning algorithms from digital geometry viewpoint. *Fourth International Joint Conference on Pattern Recognition*, Kyoto, 715-719.

[39] Tsao, Y.F., and K.S. Fu (1981). Parallel thinning operations for digital binary images. *Proceedings of PRIP'81.*

[40] Tsao, Y.F., and K.S. Fu (1981). A parallel thinning algorithm for 3-D pictures. *Computer Graphics and Image Processing* 17, 315-331.

[41] Wakayama, T. (1982). A core-line tracing algorithm based on maximal square moving. *IEEE Trans. Pattern Anal. Machine Intell.* 4, 68-74.

[42] Wang, P.S.P., and Y.Y. Zhang (1989). A fast and flexible thinning algorithm. *IEEE Trans. Computers* 38, 741-745.

[43] Woetzel, G. (1978). A fast and economic scan-to-line conversion algorithm. *Computer Graphics* 12, 125-129.

[44] Xia, Y. (1989). Skeletonization via the realization of the fire front's propagation and extinction in digitial binary shapes. *IEEE Trans. Pattern Anal. Machine Intell.* 11, 1076-1086.

[45] Zhang, T.Y., and C.Y. Suen (1984). A fast parallel algorithm for thinning digital patterns. *Comm. ACM* 27, 236-239.