between two crossing convex polygons," *Computing*, vol. 32, 1984, pp. 357-364.

[To85a] Toussaint, G. T., ed., *Computational Geometry*, North-Holland, 1985.

[To85b] Toussaint, G. T., "New results in computational geometry relevant to pattern recognition in practice," in *Proc. Pattern Recognition in Practice II*, Amsterdam, June 19-21, 1985.

[To85c] Toussaint, G. T., "On the complexity of approximating polygonal curves in the plane," *Proc. IASTED International Symposium on Robotics & Automation*, Lugano, Switzerland, 1985.

[To86] Toussaint, G. T. "Computational geometry and morphology," *Science on Form: Proc. First International Symposium for Science on Form*, S. Ishizaka, Ed., KTB Scientific Publishers, Tokyo, 1986, pp. 395-403.

[To88a] Toussaint, G. T., ed., *Computational Morphology*, North-Holland, 1988.

[To88b] Toussaint, G. T., ed., *The Visual Computer*, Special Issue on Computational Geometry, vol. 3, No. 6, May 1988.

[To88c] Toussaint, G. T., "A graph-theoretical primal sketch," in *Computational Morphology*, Toussaint, G. T., ed., North-Holland, 1988, pp. 229-260.

[To88d] Toussaint, G. T., "Computing visibility properties of polygons," in *Pattern Recognition & Artificial Intelligence*, E.S. Gelsema & L. N. Kanal, Eds., North Holland, 1988, pp. 103-122.

[To89] Toussaint, G. T., "Computing geodesic properties inside a simple polygon," *Revue d'Intelligence Artificielle*, vol. 3, No. 2, 1989, pp. 9-42.

[To92a] Toussaint, G. T., ed., *Pattern Recognition Letters*, Special Issue on Computational Geometry, to appear in 1991.

[To92b] Toussaint, G. T., ed., *Proceedings of the IEEE*, Special Issue on Computational Geometry, to appear in 1992.

[Ur83] Urquhart, R. B. "Some properties of the planar Euclidean relative neighborhood graph," *Pattern Recognition Letters*, vol. 1, July 1983, pp. 317-322.

[Wa73] Wagner, T. J., "Deleted estimates of the Bayes risk," *Annals of Statistics*, vol. 1, March 1973, pp. 359-362.

[We78] Wezka, J. S., "A survey of threshold selection techniques," *Computer Graphics and Image Processing*, vol. 7, 1978, pp. 259-265.

[Yi93] Yianilos, P. N., "Data structures and algorithms for nearest neighbor search in general metric spaces," *Symposium on Data Structures and Algorithms*, Austin, Texas, January 1993, pp. 311-321.

lbaum, 1987.

[TA82] Toussaint, G. T. and Avis, D., "On a convex hull algorithm for polygons and its application to triangulation problems," *Pattern Recognition*, vol. 15, 1982, pp. 23-29.

[TA92] Tsujimoto, S. and Asada, H., "Major components of a complete text reading system," *Proceedings of the IEEE*, vol. 80, No. 7, July 1992, pp. 1133-1149.

[TB81] Toussaint, G. T., and Bhattacharya, B. K., "Optimal algorithms for computing the minimum distance between two finite planar sets," *Proc. Fifth International Congress of Cybernetics and Systems*, Mexico City, August 1981.

[TBP84] Toussaint, G. T., Bhattacharya, B. K., and Poulsen, R. S., "The application of Voronoi diagrams to non-parametric decision rules," *Proc. Computer Science & Statistics: 16th Symposium on the Interface*, Atlanta, Georgia, March 14-16, 1984.

[TM82] Toussaint, G. T., and McAlear, J. A., "A simple O(n log n) algorithm for finding the maximum distance between two finite planar sets," *Pattern Recognition Letters*, vol. 1, October 1982, pp. 21-24.

[To70] Toussaint, G. T., "On a simple Minkowski metric classifier," *IEEE Transactions on Systems Science & Cybernetics*, vol. SSC-6, October 1970, pp. 360-362.

[To72] Toussaint, G. T., "Polynomial representation of classifiers with independent discrete-valued features," *IEEE Transactions on Computers*, vol. C-21, February 1972, pp. 205-208.

[To74] Toussaint, G. T., "Bibliography on estimation of misclassification," *IEEE Transactions on Information Theory*, vol. IT-20, July 1974, pp. 472-479.

[To78] Toussaint, G. T., "The use of context in pattern recognition," *Pattern Recognition*, vol. 10, 1978, pp. 189-204.

[To80a] Toussaint, G. T. "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, 1980, pp. 261-268.

[To80b] Toussaint, G. T. "Decomposing a simple polygon with the relative neighborhood graph," *Proceedings of the Allerton Conference*, October 1980, pp. 20-28.

[To80c] Toussaint, G. T., "Pattern recognition and geometrical complexity," *Proc. Fifth International Joint Conf. on Pattern Recognition*, Miami, December 1980.

[To83] Toussaint, G. T., "On the application of the convex hull to histogram analysis in threshold selection," *Pattern Recognition Letters*, vol. 2, Dec. 1983, pp. 75-77.

[To83b] Toussaint, G. T., "Solving geometric problems with the rotating calipers," *Proc. MELECON'83*, Athens, Greece, May 1983.

[To84] Toussaint, G. T., "An optimal algorithm for computing the minimum vertex distance and implementation methodology," *Proc. Second Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, U.S.A., April 26-28, 1993, pp. 65-104.

[PH74] Pavlidis, T. and Horowitz, S. L., "Segmentation of plane curves," *IEEE Transactions on Computers*, vol. C-23, August 1974, pp. 860-870.

[Pr83] Preparata, F., ed., *Computational Geometry*, JAI Press, 1983.

[PS85] Preparata, F. P. and Shamos, M. I., *Computational Geometry,* Springer-Verlag, 1985.

[Ra88] Radke, J. D. "On the shape of a set of points," in *Computational Morphology,* Toussaint, G. T., ed., North-Holland, 1988, pp. 105-136.

[Ro93] Robert, J.-M., "Maximum distance for two sets of points in E^d," *Pattern Recognition Letters*, vol. 14, September 1993.

[Ro69] Rosenfeld, A., *Picture Processing by Computer*, Academic Press, 1969.

[RS86] Rastogy, A. and Srihari, S. N., "Recognizing textual blocks in document images using the Hough transform," Report Tr-86-01, Department of Computer Science, SUNY at Buffalo, 1986.

[RT83] Rosenfeld, A. and de la Torre, P., "Histogram con-cavity analysis as an aid in threshold selection," *IEEE Trans. Systems, Man and Cybernetics*, vol. 13, 1983, pp. 231-235.

[Ru75] Rutovitz, D., "An algorithm for in-line generation of a convex cover," *Computer Graphics and Image Processing*, vol. 4, 1975, pp. 74-78.

[Se82] Serra, J., *Image Analysis & Mathematical Morphology,* Academic Press, 1982.

[SF88] Senechal, M. and Fleck, G., eds., *Shaping Space: A Polyhedral Approach,* Birkhauser, 1988.

[Sr93] Srihari, S. N., "From pixels to paragraphs: the use of models in text recognition," *Proc. Second Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, U.S.A., April 26-28, 1993, pp. 47-64.

[SSH87] Schwartz, J. T., Sharir, M., and Hopcroft, J., *Planning, Geometry, and the Complexity of Robot Motion,* Norwood, 1987.

[St91] Stolfi, J., *Oriented Projective Geometry*, Academic Press, Inc., 1991.

[Su84] Sugihara, K., "An O(*n* log *n*) algorithm for determining the congruity of polyhedra," *Journal of Computer and Systems Sciences*, vol. 29., 1984, pp. 36-47.

[Su83] Supowit, K. J. "The relative neighborhood graph, with an application to minimum spanning trees," *Journal of the ACM,* vol. 30, No. 3, July 1983, pp. 428-448.

[Su86] Sugihara, K., *Machine Interpretation of Line Drawings*, M.I.T. Press, 1986.

[Su92] Sugihara, K., *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams,* John Wiley and Sons, London, 1992.

[SY87] Schwartz, J. T. and Yap, C. K., *Algorithmic and Geometric Aspects of Robotics*, Er-

puting, vol. 14, No. 4, November 1985, pp. 799-817.

[Kl89] Klein, R., *Concrete and Abstract Voronoi Diagrams*, Springer-Verlag, 1989.

[Kn92] Knuth, D. E., *Axioms and Hulls*, Springer-Verlag, 1992.

[Le82] Lee, D. T. "Medial axis transformation of a planar shape," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-4, No. 4, July 1982, pp. 363-369.

[LP77] Lee, D. T., and Preparata, F. P., "Location of a point in a planar subdivision and its applications," *SIAM Journal of Computing*, vol. 6., 1977, pp. 594-606.

[LT87] Leou, J.-J. and Tsai, W.-H., "Automatic rotational symmetry determination for shape analysis," *Pattern Recognition*, vol. 20, No. 6, 1987, pp. 571-582.

[LTW94] Le, D. S., Thoma, G. R. and Wechsler, H., "Automated page orientation and skew angle detection for binary document images," *Pattern Recognition*, vol. 27, No. 10, 1994, pp. 1325-1344.

[Ma72] Maruyama, K., "A study of visual shape perception," University of Illinois, Tech. Rept., UIUCDCS-R-72-533, 1972.

[Mc92] McLachlan, G. J., *Discriminant Analysis and Statistical Pattern Recognition*, John-Wiley & Sons, Inc., New York, 1992.

[Meh84] Mehlhorn, K., *Multidimensional Searching and Computational Geometry*, Springer-Verlag, 1984.

[Me87] Melkman, A., "On-line construction of the convex hull of a simple polyline," *Information Processing Letters*, vol. 25, April 1987, pp. 11-12.

[MP69] Minsky, M. and S. Papert, *Perceptrons: An Introduction to Computational Geometry*, M.I.T. Press, 1969.

[MS80] Matula D. W. and R. R. Sokal, "Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane," *Geographical Analysis*, vol. 12, 1980, pp. 205-222.

[NT70] Nagy, G., and Tuong, N., "Normalization techniques for handprinted numerals," *Communications of the ACM*, vol. 13, No. 8, August 1970, pp. 475-485.

[OIM84] Ohya, T., Iri, M. and Murota, K., "A fast Voronoi diagram algorithm with quaternary tree bucketing," *Information Processing Letters*, vol. 18, No. 4, 1984, pp. 227-231.

[Ol89] Olariu, S. "A simple linear-time algorithm for computing the RNG and MST of unimodal polygons," *Information Processing Letters*, vol. 31, June 1989, pp. 243-247.

[O'R87] O'Rourke, J., *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.

[PA75] Pavlidis, T. and Ali, F., "Computer recognition of hand written numerals by polygonal approximation," *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-5, November 1975, pp. 610-614.

[PCHH93] Phillips, I. T., Chen, S., Ha, J. and Haralick, R. M., "English document database design pp. 193-233.

[GB78] Getis A. and B. Boots, *Models of Spatial Processes: An Approach to the Study of Point, Line, and Area Patterns*, Cambridge University Press, 1978.

[GS78] Green, P. J. and Sibson, R., "Computing Dirichlet tessellations in the plane," *The Computer Journal*, vol. 21, 1978. pp. 168-173.

[HFD90] Hinds, S., Fisher, J. and D'Amato, D., "A document skew detection method using run-length encoding and the Hough transform," *Proc. 10th International Conference on Pattern Recognition*, 1990, pp. 464-468.

[HH89] Hopcroft, J. E., and Huttenlocher, D. P., "On planar point matching under affine transformation," *First Canadian Conference on Computational Geometry*, McGill University, August 1989, also Tech. Rept. 89-986, Dept. Computer Science, Cornell University, April 1989.

[Ho86] Horn, B. K. P., *Robot Vision*, M.I.T. Press, 1986.

[HS85] Haralick, R. M., and Shapiro, L. G., "Survey-image segmentation techniques," *Computer Vision, Graphics & Image Processing*, vol. 29, 1985, pp. 100-132.

[HS89] Hershberger, J. and Suri, S., "Finding tailored partitions," *Proc. Fifth ACM Symposium on Computational Geometry*, Saarbruchen, June 5-7, 1989, pp. 255-265.

[HT73] Hopcroft, J. E. and Tarjan, R. E., "A V log V algorithm for isomorphism of triconnected planar graphs," *Journal of Computer and System Sciences*, vol. 7, 1973, pp. 323-331.

[HU87] Huttenlocher, D. P., and Ullman, S., "Object recognition using alignment," *Proc. First International Conf. on Computer Vision*, IEEE Computer Society Press, 1987, pp. 102-111.

[II85] Imai, H., and Iri, M., "Computational geometric methods for polygonal approximations of a curve," Tech. Rept. RMI 85-01, January 1985, University of Tokyo.

[II88] Imai, H., and Iri, M., "Polygonal approximations of a curve - Formulations & algorithms," in *Computational Morphology*, G. T. Toussaint, Ed., North-Holland, 1988.

[It93] Ittner, D. J., "Automatic inference of textline orientation," *Proc. Second Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, U.S.A., April 26-28, 1993, pp. 123-133.

[JS71] Jardine, N. and Sibson, R., *Mathematical Taxonomy*, John Wiley, 1971.

[JT92] Jaromczyk, J. W. and Toussaint, G. T., "Relative neighborhood graphs and their relatives," *Proceedings of the IEEE*, vol. 80, No. 9, September 1992, pp. 1502-1517.

[[Ke85] Keil, J. M., "Decomposing a polygon into simpler components," *SIAM Journal of Com-*

January 1981, pp. 75-78.

[De86] Dehne, F., *Parallel Computational Geometry and Clustering Methods*, Wurzburg 1986.

[De85] Devroye, L. "Expected time analysis of algorithms in computational geometry," in *Computational Geometry*, Ed., G. T. Toussaint, North-Holland, 1985, pp. 135-151.

[De88] Devroye, L. "The expected size of some graphs in computational geometry," *Computers & Mathematics with Applications,* vol. 15, No. 1, 1988, pp. 53-64.

[DF90] Dehne, F. and Ficocelli, L., "An efficient computational geometry method for detecting dotted lines in noisy images," *The Computer Journal*, vol. 33, No. 5, 190, pp. 424-428.

[DH72] Duda, R. O. and Hart, P. E., *Pattern Classification & Scene Analysis*, John-Wiley & Sons, 1972.

[DK81] Devroye, L. and Klincsek, T. "Average time behavior of distributive sorting algorithms," *Computing*, vol. 26, 1981, pp. 1-7.

[DSS92] Dickerson, M. T., Scott Drysdale, R. L. and Sack, J.-R., "Simple algorithms for enumerating interpoint distances and finding *k* nearest neighbors," *International Journal of Computational Geometry and Applications*, vol. 2, No. 3, 1992, pp. 221-239.

[Ea88] Eades, P., "Symmetry finding algorithms," in *Computational Morphology*, ed., G. T. Toussaint, North-Holland, 1988, pp. 41-51.

[Ed87] Edelsbrunner, H., *Algorithms in Combinatorial Geometry*, Springer-Verlag, 1987.

[ET88] ElGindy H. A. and Toussaint, G. T., "Computing the relative neighbor decomposition of a simple polygon," in *Computational Morphology,* G. T. Toussaint, Editor, North-Holland, pp. 53-70.

[ET94] Eu, D. and Toussaint, G. T., "On approximating polygonal curves in two and three dimensions," *Graphical Models and Image Processing*, to appear in 1994.

[FBF77] Friedman, J. H., Bentley, J. L., & Finkel, R. A., "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software*, vol. 3, September 1977, pp. 209-226.

[FNK92] Fujisawa, H., Nakano, Y. and Kurino, K., "Segmentation methods for character recognition: from segmentation to document analysis," *Proceedings of the IEEE*, vol. 80, No. 7, July 1992, pp. 1079-1092.

[FP75] Feng, H.-Y., and T. Pavlidis, "Decomposition of polygons into simpler components: Feature generation for syntactic pattern recognition," *IEEE Trans. Computers,* vol. C-24, No. 6, June 1975, pp. 636-650.

[Fo87] Fortune, S., "Sweepline algorithms for Voronoi diagrams," *Algorithmica*, vol. 2, 1987, pp. 153-174.

[Fo92] Fortune, S., "Voronoi diagrams and Delaunay triangulations," in *Computing in Euclidean Geometry*, D.-Z. Du and F. K. Hwang, Eds., World Scientific Publishing Co., 1992, *Scientists and Engineers*, May 1987.

[Ba90] Baird, H., "Background structure in document images," *Proc. IAPR Workshop on Structural and Syntactic Pattern Recognition*, August 1992.

[Ba92] Baird, H., "Anatomy of a versatile page reader," *Proceedings of the IEEE*, vol. 80, No. 7, July 1992, pp. 1059-1065.

[BB82] Ballard, D. H. and Brown, C. M., *Computer Vision*, Prentice-Hall, Inc., 1982.

[BJF90] Baird, H., Jones, S. and Fortune, S., "Image segmentation using shape-directed covers," *Proc. IAPR 10th International Conference on Pattern Recognition*, June 1990.

[BT83] Bhattacharya, B. K., and Toussaint, G. T., "Efficient algorithms for computing the maximum distance between two finite planar sets," *Journal of Algorithms,* vol. 4, 1983, pp. 121-136.

[BT83b] Bhattacharya, B. K., and Toussaint, G. T., "Time-and storage-efficient implementation of an optimal planar convex hull algorithm," *Image and Vision Computing,* vol. 1, No. 3, August 1983, pp. 140-144.

[BT87] Bhattacharya, B. K., and Toussaint, G. T., "Fast algorithms for computing the diameter of a finite planar set," *Proc. Computer Graphics International 1987,* Ed., T. L. Kunii, Springer-Verlag 1987, pp. 89-104.

[BWY80] Bentley, J., Weide, B. and Yao, A., "Optimal expected-time algorithms for closest point problems," *ACM Transactions of Mathematical Software*, vol. 6, 1980, pp. 563-580.

[Bo92] Bokser, M., "Omnidocument technologies,"*Proceedings of the IEEE,* vol. 80, No. 7, July 1992, pp. 1066-1078.

[CC92] Chan W. S. and Chin, F., "Approximation of polygonal curves with minimum number of line segments," *Proc. 2nd SAAC'92,* December 1992, Nagoya, Japan, pp. 378-387.

[Ch75] Chvatal, V., "A combinatorial theorem in plane geometry," *Journal of Combinatorial Theory Series B*, vol. 18, 1975, pp. 39-41.

[CH67] Cover, T. M., and Hart, P. E., "Nearest neighbour pattern classification," *IEEE Transactions on Information Theory,* vol. IT-13, 1967, pp. 21-27.

[CPT77] Cahn, R. L., Poulsen, R. S., and Toussaint, G. T., "Segmentation of cervical cell images," *Journal of Histochemistry and Cytochemistry*, vol. 25, No. 7, 1977, pp. 681-688.

[CT77] Cohen, M. and Toussaint, G. T., "On the detection of structures in noisy pictures," *Pattern Recognition*, vol. 9, 1977, pp. 95-98.

[CT76] Cheriton, D. and Tarjan, R., "Finding minimum spanning trees," *SIAM Journal of Computing*, vol. 5, No. 4, December 1976.

[De81] Devroye, L. P., "On the inequality of Cover & Hart in nearest neighbour discrimination," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. PAMI-3,

proper experimental methodology for estimating the performance of a decision rule. For a survey of early work on this topic see [To74]. For the latest results see the complete and thorough text by McLachlan [Mc92]. Many geometric problems occur here as well where computational geometry offers solutions. For example, a good method of estimating the performance of the NN-rule is to delete each member of $\{\mathbf{X},\Theta\} = \{(\mathbf{X}_1,\theta_1), (\mathbf{X}_2,\theta_2),..., (\mathbf{X}_n,\theta_n)\}$ in turn and classify it with the remaining set [Wa73]. Geometrically this problem reduces to computing for a given set of points in *d*-space the nearest neighbour of each (the all-nearest-neighbours problem). For the latest and most practical computational geometric results concerning nearest neighbour search in all dimensions and for pointers to many other key recent results the reader is referred to [DSS92], [Yi93] and [AM93].

6. References

[ABGW90] Alt, H., Blomer, J., Godau, M. and Wagener, H., "Approximation of convex polygons," *ICALP*, 1990.

[AEIIM85] Asano, T., Edahiro, M., Imai, H., Iri, M. and Murota, K., "Practical use of bucketing techniques in computational geometry," in *Computational Geometry*, Ed., G. T. Toussaint, North-Holland, 1985, pp. 153-195.

[AGSS] Aggarwal, A., L. Guibas, J. Saxe, and P. W. Shor, "A linear time algorithm for computing the Voronoi diagram of a convex polygon," *Proc. 19th ACM Symposium on the Theory of Computing,* 1987, pp. 39-45.

[AH85] Avis D. and J. Horton, "Remarks on the sphere of influence graph," in *Discrete Geometry and Convexity,* Eds., J. E. Goodman et al., New York Academy of Sciences, 1985, pp. 323-327.

[AM93] Arya, S. and Mount, D. M., "Approximate nearest neighbor queries in fixed dimensions," *Symposium on Data Structures and Algorithms*, Austin, Texas, January 1993, pp. 271-280.

[AMWW88] Alt, H., Mehlhorn, K., Wagener, H. and Welzl, E., "Congruence, similarity and symmetries of geometric objects," *Discrete & Computational Geometry,* vol. 3, 1988, pp. 237-256.

[ART87] Aggarwal, A., P. Raghavan, and P. Tiwari, "Lower bounds for closest pair and related problems in simple polygons," *IBM T. J. Watson Tech. Rept.,* in press.

[AS83] Ahuja, N., and Schacter, B. J., *Pattern Models,* John Wiley, 1983.

[AT78] Akl, S. G. and Toussaint, G. T., "An improved algorithm to check for polygon similarity," *Information Processing Letters*, vol. 7, 1978, pp. 127-128.

[AT81] Avis, D. and Toussaint, G. T., "An efficient algorithm for decomposing a polygon into star-shaped pieces," *Pattern Recognition,* vol. 13, 1981, pp. 295-298.

[Ba68] Bartz, M. R., "The IBM 1975 optical page reader," *IBM J. Res. Develop.,* vol. 12, September 1968, pp. 354-363.

[Ba87] Baird, H., "The skew angle of printed documents," *Proc. Conf. Society of Photographic*

[Ol89].

5. Decision Rules

Once a feature vector $\boldsymbol{X}=[x_1,x_2,...,x_d]$ has been extracted from an object in the image it is often desired to classify the object into one of a predetermined set of pattern classes or categories. There are scores of methods for doing this [DH72].

5.1 Parametric Decision Rules

In parametric classification we assume that $\boldsymbol{X}$ is a random variable with some specified probability density function or distribution described by some parameters that are usually estimated from data. In this approach one is often called upon to compute distances between sets under varying types of metrics [To70]. Such is in fact an implicit computation of the Voronoi diagram where the seeds are the estimates of location for the distributions. Alternately, one may seek to describe geometrically the decision boundaries themselves, i.e., the manner in which the discriminant functions partition the feature space into regions associated with the pattern classes [To72].

5.2 Non-parametric Decision Rules

In the non-parametric classification problem we have available a set of n feature vectors taken from a collected data set of n objects denoted by $\{\mathbf{X},\Theta\}=\{(\mathbf{X}_1,\theta_1), (\mathbf{X}_2,\theta_2),..., (\mathbf{X}_n,\theta_n)\}$, where $\mathbf{X}_i$ and θ_i denote, respectively, the feature vector on the ith object and the class label of the object. One of the most powerful such techniques is the so-called nearest-neighbour rule (NN-rule) [CH67], [De81]. Let $\mathbf{Y}$ be a new object (feature vector) to be classified and let $\mathbf{X}_k^* \in \{\mathbf{X}_1,\mathbf{X}_2,...,\mathbf{X}_n\}$ be the feature vector closest to $\mathbf{Y}$. The nearest neighbour decision rule classifies the unknown object $\mathbf{Y}$ as belonging to class θ_k^*.

In the 1960's and 1970's pattern recognition practitioners have avoided using the NN-rule on the grounds of the mistaken assumptions that (1) all the data $\{\mathbf{X},\Theta\}$ must be stored in order to implement such a rule and (2) to determine $\mathbf{X}_k^*$, distances must be computed between $\mathbf{Y}$ and all members of $\{\mathbf{X}_1,\mathbf{X}_2,...,\mathbf{X}_n\}$. Computational geometric progress in the 1980's and 1990's has made the NN-rule a practical reality. Indeed, it is recommended as the decision rule of choice in practice. It is surprising that in the OCR literature the misconceptions concerning the NN-rule are still present. For example, Bokser [Bo92] states concerning the memory requirements of the nearest neighbor classifier that *"if the classifier requires a library of 200,000 vectors to achieve acceptable accuracy on the training set, then 200,000 distances must be computed at run time to classify each input vector."* Nothing could be further from the truth. Both of the above problems have been eradicated with techniques from computational geometry. Various methods exist for computing a nearest neighbour without computing distances to all the candidates [FBF77]. In fact, the point location techniques [LP77] do not compute any distances at all. Furthermore, not all the "training" data $\{\mathbf{X},\Theta\}$ is required to be stored. Methods have been developed [TBP84] to edit "redundant" members of $\{\mathbf{X},\Theta\}$ in order to obtain a relatively small subset of $\{\mathbf{X},\Theta\}$ that nevertheless implements exactly the same decision rule as using all of $\{\mathbf{X},\Theta\}$. Such methods depend heavily on the use of Voronoi diagrams and proximity graphs such as the Gabriel graph [TBP84].

5.3 Estimation of Misclassification

A most important and still too often neglected problem in pattern recognition concerns the can be computed efficiently in O(n log n) time. For a survey of the most recent results in this area the reader is referred to the paper by Radke [Ra88]. It is expected that most of these proximity graphs can find a place in the document analysis problem where they can make practical contributions.

4.3 Polygon Decomposition

4.3.1 Simple polygons

In 1975 Vasek Chvatal [Ch75] proved that n/3 guards were always sufficient, and sometimes necessary, to guard (jointly see) the complete interior of a simple polygon (art gallery) consisting of n walls or vertices. This result has come to be known as Chvatal's *Art Gallery Theorem* and has since evolved to fill out an entire book on the subject [O'R 87]. Avis and Toussaint in 1981 obtained an O(n log n) time algorithm for actually placing the guards and noted that this algorithm also decomposes the polygon into at most n/3 star-shaped components [AT81] improving on the complexity of a previous algorithm for this problem [Ma72].

The problems of decomposing simple polygons into various types of more structured polygons have a number of practical applications and have received considerable attention recently from the theoretical perspective. See [To88a] for several papers discussing recent issues. We have already seen in the section on text-block isolation in the page-segmentation problem that decomposition of the white empty spaces into maximal convex components provides an approach to that problem. In character recognition, on the other hand, it is desired to obtain decompositions of a simple polygon into perceptually meaningful parts. The so-called *component-directed* methods or *region-based* covers and partitions decompose the polygon into well established classes of simpler polygons such as triangles, squares, rectangles as well as convex, monotone, or star-shaped polygons [To88a]. These decompositions however are rarely satisfactory from the morphological point of view although they do have their place in other contexts. An alternate approach which may be superior from this point of view is the use of *procedure-directed* methods based on proximity graphs. In [To80b] it was proposed to use the *relative-neighbour decomposition* (RND) of a simple polygon P of n vertices and an O(n^3) time algorithm for its computation was given. ElGindy and Toussaint [ET88] have since reduced this complexity to O(n^2). Two vertices p_i and p_j of a simple polygon are relative neighbours if their lune contains no other vertices of P that are visible from either p_i or p_j. Two vertices p_i and p_j are *visible* if the line segment [p_i, p_j] lies in P.

4.3.2 Special classes of polygons

The fastest known algorithm [ET88] for computing the RND of a simple polygon is O(n^2). On the other hand, for *convex* polygons the RND can be computed in O(n) time [Su83], and so can the Delaunay triangulation [AGSS]. However, it is shown in [ART87] that Ω(n log n) is a lower bound for computing the Delaunay triangulation on the vertices of a *star-shaped* or *monotone* polygon. It is unknown whether any other proximity graphs can be computed in linear time for the case of convex polygons. Furthermore, for most proximity graphs it is unknown whether they can be computed in o(n^2) time for special classes of simple polygons such as *star-shaped*, *monotone* or *unimodal* polygons. For *unimodal* polygons the RNG and MST can be computed in O(n) time ture vector $\boldsymbol{X}=[x_1,x_2,...,x_d]$. Thus an object, modeled as a polygon P, is mapped through this process into a point in d-dimensional *feature-space*. Most features employed are of a geometric nature and computational geometry has much to contribute to this aspect of OCR as well. For example the medial axis of P is a very powerful morphological descriptor [Le82] as are visibility [To88d] and geodesic [To89] properties.

Symmetry is an important feature in the analysis and synthesis of shape and form [LT87]. As such it is not surprising that it has received considerable attention in the pattern recognition, image processing, and computer graphics literatures. One of the earliest applications of computational geometry to symmetry detection was the algorithm of Akl & Toussaint [AT78] to check for polygon similarity. Since then attention has been given to other aspects of symmetry and for objects other than polygons. For example, Sugihara [Su84] shows how a modification of the planar graph-isomorphism algorithm of Hopcroft and Tarjan [HT73] can be used to find all symmetries of a wide class of polyhedra in O(n log n) time. For a survey of the most recent work on detecting symmetry see [Ea88].

4.2 The Shape of a Set of Points

In some contexts such as (1) the analysis of pictures of bubble-chamber events in particle physics and (2) textline inference in document analysis from the center points of the black rectangles of connected components, the input patterns are not well described by polygons because the patterns consist of a set of disconnected dots. Such "objects" are called *dot-patterns* and are well modeled as sets of points. Thus one of the central problems in shape analysis is extracting or describing the shape of a set of points. Let $S=\{x_1, x_2,..., x_n\}$ be a finite set of points in the plane. A *proximity graph* on a set of points is a graph obtained by connecting two points in the set by an edge if the two points are close, in some sense, to each other. The minimal spanning tree (MST) [CT76], the relative neighborhood graph (RNG) [To80a] and the β-skeletons [JT92] are three proximity graphs that have been well investigated in this context. The lune of x_i and x_j, denoted by Lune(x_i, x_j), is defined as the intersection of the two discs centered at x_i and x_j with radius equal to the distance between x_i and x_j. The RNG is obtained by joining two points x_i and x_j of S with an edge if Lune(x_i, x_j) does not contain any other points of S in its interior. By generalizing the shape of Lune(x_i, x_j) one obtains generalizations of the RNG. One of the best known proximity graphs on a set of points is the Delaunay triangulation (DT) and it is well known that the RNG is a supergraph of the MST and the DT is a supergraph of the RNG [To80a]. The β-skeletons are a generalization of RNG's and Gabriel graphs [MS80] and the lune-based neighborhoods in question are a function of a parameter β. For particular values of β, the β-skeleton reduces to the RNG and the Gabriel graph. In [To88c] a new graph termed the *sphere-of-influence* graph is proposed as a primal sketch intended to capture the low-level perceptual structure of visual scenes consisting of dot-patterns (point-sets). Avis and Horton [AH85] showed that the number of edges in the sphere-of-influence graph is bounded above by 29n. The best upper bound to date is 17.5. This follows from a lemma of Bateman in geometrical extrema suggested by a lemma of Besicovitch (*Geometry*, May 1951, pp. 667-675) and an observation of Kachalski. Bateman's lemma gives 18n and Kachalski's trick reduces it by 0.5. The same trick reduces Avis & Horton's bound by 0.5. David Avis has found examples that require 9n edges and conjectures that the best upper bound is in fact 9n. The fact that the *sphere-of-influence* graph contains at most a linear number of edges allows for its efficient computation. Furthermore, from the shape measurement point of view, the graph suffers from almost none of the drawbacks of previous methods and for a dot pattern consisting of n dots the number of vertices of the polygons while retaining their inherent shape using polygonal approximation methods in order to reduce the complexity of subsequent algorithms applied to the polygons. Such an approach was applied to character recognition by Pavlidis and his colleagues [PA75], [PH74]. Here again is an area where computational geometry has great potential and is playing an ever increasing role. Smoothing and enhancement can be carried out for example by deleting carefully chosen branches of the medial axis of the polygon [Le82]. Given a polygonal planar curve $P = (p_1, p_2, ..., p_n)$ the polygonal approximation problem can be cast in many different molds. One such version for example calls for determining a new curve $P' = (p'_1, p'_2, ..., p'_m)$ such that, 1) $m < n$, 2) the p'_i are a subset of the p_i, and 3) any line segment $[p'_j, p'_{j+1}]$ which substitutes the chain corresponding to $[p_r, ..., p_s]$ in P is such that the distance between every p_k for k between r and s and the approximating line segment is less than some predetermined error tolerance. Recently Iri and Imal [II85] proposed an elegant $O(n^3)$ algorithm that finds the approximation that minimizes m subject to the two other constraints. In [To85c] it is shown how the complexity of their algorithm can be reduced to $O(n^2 \log n)$ time when the error criterion is changed. Furthermore, it is shown that the complexity of the method can be further reduced to $O(n^2)$ if the curves are monotonic in a known direction. Since then these algorithms have been improved again to run in $O(n^2)$ time for arbitrary simple polygonal chains and different error criteria [ET94], [CC92].

For a still up-to-date survey of polygonal approximation techniques the reader is referred to the excellent paper by Imai & Iri [II88].

3.3 Pattern Matching

One approach to character recognition avoids feature extraction or shape analysis altogether and instead tries to match a set of points A (fiducial points obtained from the unknown object) to a pre-stored set B from a collection of sets representing the different pattern classes. The geometric problem here is to determine whether there exists an affine transformation (a general linear transformation followed by a translation) that maps each point of A onto a corresponding point of B. Only recently has computational geometry been invoked here [HU87], [HH89] and much work remains to be done. For the special case in which the cardinalities of A and B are equal, whether such a transformation exists can be determined in $\theta(n \log n)$ time where n is the said cardinality [HH89]. For a variety of computational geometric results in this area the reader is referred to [AMWW88]. A related problem here is to compute the *similarity* or *distance* between two polygons which could represent the boundaries of shapes or the convex hulls of sets of points [To84]. This problem is in turn closely related to the problem of approximating polygons by smoother ones or by polygons with fewer vertices [ABGW90], [To85a].

4. Computational Morphology and Shape Analysis

Computational morphology is concerned with the analysis, description, and synthesis of shapes and patterns from a computational point of view. It is therefore of central concern to document analysis. Once the objects in an image have been normalized, smoothed, and cleaned up it is time to measure their shape using mathematical descriptors of shape [Se82]. This is referred to as feature extraction.

4.1 Feature Extraction

Typically we calculate d features or measurements of the shape of an object yielding a feathese should run much faster than Fortune's for this problem. In fact several approaches have been developed for obtaining O(n) expected time algorithms. Bentley, Weide and Yao [BWY80] show that a combination of *bucketing* techniques and any O(n log n) *worst-case* time algorithm yields an algorithm with O(n) expected time. Devroye [De85] has proposed additional bucketing algorithms for computing the MST in O(n) expected time. Probably the best practical algorithm with the most promise for the textline inference problem is the *quaternary-incremental-algorithm* of Ohya, Iri and Murota [OIM84]. This algorithm is a modification of the more primitive incremental O(n^2) worst-case algorithm of Green and Sibson [GS78]. The *quaternary-incremental-algorithm* introduces a special bucketing technique, stored as a quaternary tree, to determine the order in which points must be inserted into the growing Voronoi diagram. An extensive experimental comparison of this algorithm with other algorithms is described in [AEIIM85] where it is established empirically that the algorithm runs in O(n) *expected* time, is faster than the other algorithms tested and for a thousand points runs in 0.25 seconds. It is conjectured that using this algorithm to compute the Delaunay triangulation in Ittner's approach will significantly speed up the resulting textline inference algorithm. The lesson to be learned here is that for this application the points are uniformly distributed and there are usually no more than, say, 2,000 points per text block.

3. Image Processing of Characters

Once the characters or objects in the image have been isolated they are massaged in one form or another with the goal of making eventual classification easier. At this stage the objects may be treated simply as a connected collection of pixels which are processed usually in parallel in the more traditional forms of image processing [MP69], [Ro69], or they may be represented by their boundary as polygons and processed using computational geometry in the more modern approach [ET88], [Ke85] which nevertheless has early roots in the pioneering work of Feng & Pavlidis [FP75].

3.1 Normalization

Normalization is performed to make feature extraction simpler and to obtain better results. Many such techniques are inherently geometric in nature. For example, in the context of handprinted numeral recognition Nagy & Tuong [NT70] compute the convex hull of the boundary polygon of a numeric character, determine its four extreme points in the diagonal directions and then use a geometric projective transformation to map the resulting quadrilateral into a square. Other approaches involve finding the minimum-area rectangle enclosing the polygon for which a very simple and practically fast linear-time algorithm is known [To83b]. Computing the convex hull of a simple polygon is a problem fundamental at many levels in the document analysis problem and here computational geometry has made great strides. In the past twenty years a score of linear time algorithms have been published. Unfortunately it is a trickier problem than it appears and a dozen of these algorithms are not correct, i.e., are not guaranteed to work for all simple polygons. Of the correct algorithms there are a variety available that have different conceptual levels of difficulty. The fastest and simplest algorithm which is recommended for practical applications in document analysis is Melkman's algorithm [Me87].

3.2 Smoothing, Enhancement & Approximation

In spite of the application of normalization and noise removal the resulting boundary polygons of objects may still require smoothing or enhancement and it may also be desired to reduce according to a universal type-setting convention guided by ease of reading, characters are printed closer together within textlines than between textlines.

One of the most successful, robust, skew-tolerant, elegant and simple techniques for text-line orientation inference was recently proposed by Ittner [It93]. His method depends fully on recent developments in computational geometry. However, the best tools from computational geometry were unfortunately not used. Here we demonstrate how the *expected* complexity tools of computational geometry can be used to improve Ittner's method even further. We first briefly sketch Ittner's method. For details the reader is referred to [It93]. To be more precise assume that the given text block B consists of n black connected components (characters). The three key steps in the procedure are (1) idealize each character by a point, thus obtaining a set S of n points in the plane, (2) construct the Euclidean minimal spanning tree of S (MST(S)) of the n points obtained in (1), and (3) determine the textline orientation by analysis of the distribution of the orientations of the edges in the MST(S). Step (1) is done by computing the center of the bounding box of each character and does not concern us here. Computational geometry is used to solve step (2) in two phases. Cheriton and Tarjan [CT76] proposed a simple algorithm for computing the MST of a graph in O(E) time where E is the number of edges in the graph. If we join every point of S to every other we can certainly use the Cheriton-Tarjan algorithm on the resulting *complete* graph. However such a graph has $E = n(n\text{-}1)/2$ edges and hence the MST algorithm would run in O(n^2) time. Fortunately there are many graphs defined on S (usually belonging to the class of *proximity* graphs [JT92]) that have the property that they contain the MST(S) and also have O(n) edges. For these graphs the Cheriton-Tarjan algorithm runs only in O(n) time. One such graph is the dual of the well known Voronoi diagram of S. This graph is usually called the Delaunay triangulation. Ittner [It93] proposes computing the Delaunay triangulation with Fortune's sweep-line algorithm [Fo87]. Fortune's algorithm runs in O(n log n) time and is one of the most elegant and simple algorithms for computing the Delaunay triangulation of a set of points *that* runs in O(n log n) time. However there are simpler algorithms that should run much faster on the textline inference problem, for the reasons described below.

We turn therefore to how Ittner's method can be improved further. The weakness in Ittner's implementation of his novel and elegant ideas lies in the failure to distinguish between the *worst-case* complexity (running time) of algorithms and the *expected* complexity. On any given input an algorithm takes a certain amount of time. On some inputs it runs faster and on others slower. The input that makes the algorithm take the longest amount of time is the *worst-case* complexity. The input that makes the algorithm take the shortest amount of time is the *best-case* complexity. The time an algorithm takes on an *average* type of input is the *expected* complexity. Fortune's algorithm runs in O(n log n) *worst-case* time. However, it also runs in O(n log n) *expected* time. Furthermore, while it is simple compared to many other O(n log n) *worst-case* time algorithms, it is not as simple as some existing O(n^2) worst-case time algorithms because it uses non-trivial data structures such as balanced binary trees and heaps [Fo92].

The textline inference problem has a very important property that can be exploited here and that is that the characters are very much uniformly distributed in a text block. In other words the n points in S are uniformly distributed in a rectangle. This is indeed a very special type of input and for this type of input (in fact the type that is most often assumed in theory to prove the expected linearity of an algorithm!) there exist much simpler Delaunay triangulation algorithms than Fortune's which run in O(n) *expected* time and do not involve complicated data structures. Some of

[Ba92]. In particular, in the elegant method of Baird, Jones and Fortune [BJF90] the problem is solved by enumerating all *maximal* white rectangles implied by the black rectangles. A white rectangle is called maximal if it cannot be enlarged while remaining outside the black rectangles. Their enumeration algorithm takes $O(n \log n + m)$ time, where m is the number of maximal rectangles generated in the search. In the worst case $m = O(n^2)$. However using a clever heuristic to exploit properties of layouts on the average they generate only $O(n)$ maximal white rectangles. Thus they conclude that apart from the sort (for which it appears they use an $O(n \log n)$ expected time algorithm) the rest of their algorithm runs in $O(n)$ expected time. We should add here that their entire procedure can be speeded up to run in $O(n)$ time by using the fastest sorting algorithms in practice, namely the "bucket" sorting or "distributive" sorting algorithms [De85], [DK81]. Devroye [De85] has shown that for a wide class of distributions of the points, "bucket" sorting can be done in $O(n)$ expected time. The distributions of the black rectangles in the text block are essentially uniform and perfectly "tailor" made for these type of sorting algorithms. Therefore the algorithm of Baird, Jones and Fortune [BJF90] can be made to run faster by using one of these linear expected-time sorting algorithms in their first step.

2.3 Skew Determination

Once a block of text is located in the page the text-block is usually corrected for skew in order to simplify the textline determination. Variants of the Hough transform are commonly used for detecting and correcting for skew [LTW94], [HFD90], [Ba87], [RS86]. For example, in [LTW94] the image containing a block of text is processed by deleting from every connected component all black pixels other than the ones with the lowest y-coordinate. Then the Hough transform is applied to the resulting image to detect lines. The Hough transform is a rather time consuming brute-force method for mapping the original points in the *primal* space to sinusoidal curves in the *dual* space. It was first applied to detecting subsets of colinear points (line detection) in a crude way by Duda and Hart [DH72]. Later Cohen and Toussaint [CT77] showed how Duda and Hart's version could be greatly improved by taking into account the distribution of the points along values of constant angle and thus obtaining an optimal algorithm from the statistical detection theory point of view. Since then many other improvements and variations have been proposed. This is an area where computational geometry can make immediate gains. There are several computational geometric methods available for detecting colinear points which are more efficient than the Hough transform. For example, Dehne and Ficocelli [DF90] propose an $O(n \log n)$ time and $O(n)$ space algorithm for detecting dotted lines in a noisy image consisting of n dots (points). Their algorithm is based on "peeling" or successively "removing" points on the convex hull of the set. Therefore computing the convex hull of a planar set of n points is also a fundamental problem with many possible applications to document analysis. This problem is slightly more difficult than computing the convex hull of a simple or monotone polygon, two problems we encountered earlier. However, much work has been done on this problem in the computational geometry literature since 1972. The algorithm recommended for use in practice which appears to be the fastest to date is the Bhattacharya-Toussaint implementation of the Akl-Toussaint algorithm. A FORTRAN code of the algorithm is available in [BT83b].

2.4 Textline Orientation Inference

Given a block of text the textline orientation inference problem consists of determining the lines of text. Almost always these lines are either horizontal (as in English) or vertical (as in Chinese). The fundamental geometric property that allows this problem to be solved is the fact that simple tasks a single threshold which partitions the image into “figure” and “background” is sufficient. For an example of the application of thresholding to the segmentation of cervical cell images in the context of automated cervical cancer recognition the reader is referred to [CPT77]. In this example the pixels are classified into three categories corresponding to the labels: nucleus, cytoplasm, and background. Of direct relevance to document analysis, another area where thresholding is used quite successfully is character recognition [Ba68]. There are a variety of methods for selecting thresholds [We78] and computational geometry is only beginning to be applied here. For example, a frequently used heuristic for segmenting an image into grey-level clusters or objects is to select thresholds at the bottoms of “valleys” on the histogram of the digital image. In a novel approach Rosenfeld and de la Torre [RT83] proposed selecting the thresholds through a more involved analysis of the *convex deficiency* of the histogram. The convex deficiency is obtained by subtracting (in the set-theoretical sense) the histogram from its convex hull. In order to compute the convex hull of the histogram they propose an algorithm of Rutovitz [Ru75] which runs in worst-case time $O(n^2)$ where n is the number of grey levels. On the average the algorithm runs in time $O(nc)$ where c is the number of vertices on the convex hull found. However, as pointed out in [To83], the fact that a histogram is a very special type of polygon, namely a *monotonic* polygon allows us to compute the convex hull with a very simple $O(n)$ time algorithm [TA82].

2.1.2 Cluster Analysis

One of the most powerful approaches to image segmentation that lends itself to the application of complicated images such as those of magazine or newspaper documents that contain textured pictures and diagrams as well as text blocks is the method of clustering and this is an area where a great deal of computational geometry can be readily applied. In this approach each pixel is treated as a complicated object by associating it with a local neighborhood in $\boldsymbol{I}$. For example, we may define a 5×5 neighborhood of pixel p_{ij}, denoted by $N_5\,[p_{ij}]$, as $\{p_{mn} \mid i\text{-}2 \leq m \leq i\text{+}2, j\text{-}2 \leq n \leq j\text{+}2\}$. We next measure k properties of p_{ij} by making k measurements in $N_5\,[p_{ij}]$. Such measurements may include various moments of the intensity values (grey levels) found in $N_5\,[p_{ij}]$, etc. Thus each pixel is mapped into a point in k-dimensional *pixel-space*. Performing a cluster analysis of all the resulting $n \times n$ points in pixel-space yields the desired partitioning of the pixels into categories. For an elegant treatment of the subject of cluster analysis the reader is referred to the book by Jardine and Sibson [JS71]. A good treatment of the application of computational geometry to cluster analysis can be found in [De86]. For more recent and novel approaches to the problem of partitioning point sets see [HS89]. Many cluster analysis algorithms depend heavily on the computation of distances. Computational geometry has yielded efficient algorithms for many distance computation problems. The distance may be the diameter of a single set [BT87] or the minimum [TB81] or maximum [BT83], [TM82], [Ro93] distance between two sets. Other cluster analysis methods depend heavily on the computation of proximity graphs where computational geometry offers a great deal of help [JT92].

2.2 Text-Block Isolation

The text-block isolation problem consists of extracting from a digitized document blocks of text. By finding the enclosing rectangles around each connected component (character) and around the entire set of characters we have a well structured geometric object, namely, a rectangle with n rectangular “holes” (also called black rectangles). This problem is ideally suited to a computational geometric treatment. Indeed, computational geometry approaches for performing this task by analyzing the empty (white) spaces in the document have already appeared [BJF90], sent from the three texts mentioned above. However visibility is given a clear, excellent, and comprehensive treatment in the recent book by O'Rourke [O'R87]. Other computational geometric aspects of computer graphics are well treated by Stolfi [St91]. One of the most fundamental structures in computational geometry, and one that has a great deal of potential for contributing to the short term improvement of document analysis systems, is the Voronoi diagram and since the "birth" of computational geometry a score of variants on this structure have appeared. The books by Rolf Klein [Kl89], Kokichi Sugihara [Su92] and Donald Knuth [Kn92] are entirely devoted to this subject. There have also appeared three books which are collections of papers covering almost all aspects of computational geometry. The book edited by Preparata [Pr83] contains twelve papers on early material. More recent results can be found in the two books edited by Toussaint [To85a], [To88a] and in the robotics-oriented collections edited by Schwartz et al., [SSH87] and Schwartz & Yap [SY87]. Journals are also starting to devote special issues to computational geometry such as *The Visual Computer* [To88b], *Pattern Recognition Letters* [To92a], and *The Proceedings of the IEEE* [To92b]. Finally we mention a book which, although may not contain much on the *computational* aspects of geometry, certainly covers much material of direct interest to computer vision and also relevant to document analysis. This is the delightful book edited by Senechal & Fleck [SF88]. In addition to these books there exist three survey papers on those aspects of computational geometry of most relevance to the document analysis process [To80c], [To85b], and [To86].

2. Layout Analysis

The purpose of layout analysis is to obtain from a document, lines of characters (text) in preparation for the character recognition phase of the process. The main sub-problems in layout analysis are: (1) image segmentation in order to (2) obtain the black 8-connected components (unrecognized characters), (3) skew determination and text-block isolation and (4) textline orientation determination.

2.1 Image Segmentation

The transducer converts a light intensity array from the real world into a two dimensional array or digital image of *pixels* (picture elements) which are numbers resulting from a quantization of the original range of light intensity values into a pre-specified number of sub-ranges called *grey-levels*. In a binary picture there are only two levels and we speak of a "black-and-white" image. The image segmentation problem consists of receiving a digital image $\boldsymbol{I} = \{p_{ij} \mid 1 \leq i,j \leq n\}$, consisting of an $n \times n$ array (also viewed as a square lattice) of pixels p_{ij}, as input and producing a labelled planar subdivision of $\boldsymbol{I}$ as output. This presupposes labelling each pixel into categories. This having been done each connected component of $\boldsymbol{I}$ consisting of pixels with the same label or category corresponds to one of the regions in the subdivision. For a survey of general image segmentation techniques the reader is referred to [HS85]. For a survey of segmentation techniques geared to the document analysis problem see [FNK92]. We discuss only two methods here where computational geometry provides for elegant and efficient algorithms to solve the problems.

2.1.1 Histogram Analysis and Threshold Selection

One of the simplest methods of segmenting an image, but not a very powerful one for complicated images containing a variety of textures, is to compute a histogram of all the pixels with every intensity value and select some threshold values at the "significant" local minima of the histogram. Clearly, selecting k thresholds will yield k+1 categories of pixels. For simple pictures and

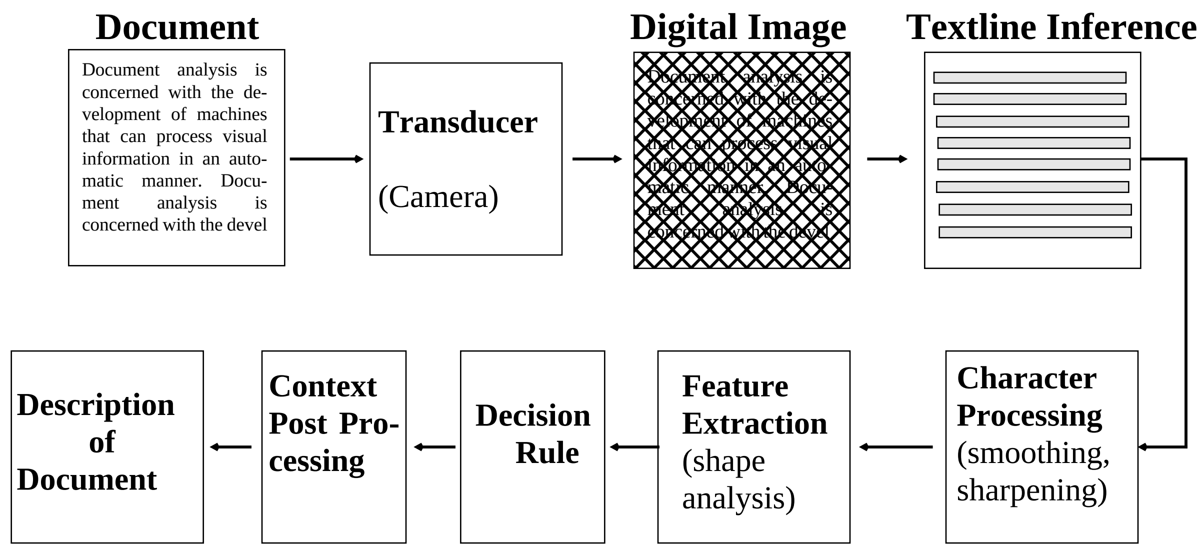

Fig. 1 Decomposing the *document analysis* problem into sub-problems.

gram is to analyze a document in the real world with the aid of an input device which is usually some form of transducer such as a digital camera and to arrive at a description of the document which is useful for the accomplishment of some task. For example, the document may consist of an envelope in the post office, the description may consist of a series of numbers supposedly accurately identifying the zip code on the envelope, and the task may be the sorting of the envelopes by geographical region for subsequent distribution. Typically the camera yields a two-dimensional array of numbers each representing the quantized amount of light or brightness of the real world scene at a particular location in the field of view. The first computational stage in the process consists of segmenting the image into meaningful objects, usually black 8-connected components. The image is also corrected for skew and shear [Ba87]. Blocks of text are then isolated usually by analyzing the structure of the white sections of the document [BJF90]. Each block of text is then analyzed to determine the orientation of the text lines which are almost always either vertical or horizontal [It93]. The next stage usually involves processing the characters to remove noise of one form or another. The third stage consists of feature extraction or measuring the "shape" of the characters. The final stage is concerned with classifying the character into one or more categories on which some subsequent task depends. This stage may also be followed or combined with algorithms for using contextual information of the language [Sr93], [To78].

Computational geometry, a twenty-year old explosive discipline of computer science, continues to flourish at an exponentially increasing rate and make its presence felt in new areas. Several books have already appeared on the subject. An introductory text by Preparata & Shamos [PS85] covers most of the early work in this area. Mehlhorn [Me84] contains a subset of the material found in Preparata & Shamos and a few different results. The combinatorial aspects of discrete and computational geometry are treated in depth in the book by Edelsbrunner [Ed87]. The question of visibility, of great interest to graphics, computer vision and robotics, is notoriously ab-

Computational Geometry for Document Analysis

Godfried T. Toussaint

School of Computer Science
McGill University
Montreal
Canada

ABSTRACT

Document analysis is concerned with the development of machines that can process visual information automatically. Computational geometry is concerned with the design of algorithms for solving geometric problems. Most problems in document analysis can be couched in geometric terms. In this paper we outline how computational geometry may significantly contribute to many aspects of the document analysis process and we provide pointers to a selection of the computational geometry literature where some of the most relevant results can be found.

1. Introduction

Document analysis is concerned with the automatic transfer by machine of visual two dimensional documents most commonly consisting of printed pages from books, magazines or newspapers [PCHH93]. Maps and engineering drawings constitute another class of common documents. The first class of problems have much in common with optical character recognition (OCR) and hence computer vision. On the other hand document analysis is a special case of computer vision and therefore its special properties give rise to special sub-problems such as text-block and textline orientation inference. Furthermore these certain special properties allow the tailoring of more general computer vision strategies resulting in better performance for certain tasks. Not surprizingly computer vision has much to offer to document analysis. For a recent collection of surveys concerning document analysis and OCR the reader is referred to the July 1992 special issue of the IEEE Proceedings.

Computer vision has flourished now for some forty years as a sub-discipline of artificial intelligence and hundreds of books are readily available on the subject and will not be mentioned here. The best early book on computer vision, and still up to date from the point of view of discriminant function analysis and Bayesian decision theory, is the text by Duda & Hart [DH73]. Popular more recent books include Ballard & Brown [BB82] and Horn [Ho86]. Finally we mention the first two books that are the fruit of the marriage between computer vision and computational geometry and these are the monographs by Ahuja & Schacter [AS83] and Sugihara [Su86].

It is useful to decompose the document analysis problem into a series of subproblems that are usually tackled sequentially and separately in some order such as that illustrated in Fig. 1. For a more complete description of the various components of the problem see [Ba92] and [TA92]. Here we concentrate on those aspects of the document analysis problem where the application of computational geometry can result in immediate gains. The purpose of a document analysis pro-