June 1990

EUROPEAN ORGANIZATION FOR EXPERIMENTAL PHOTOGRAMMETRIC RESEARCH

AUTOMATIC DIGITIZING

REPORT SUBMITTED BY A WORKING GROUP

OF

COMMISSION D (PHOTOGRAMMETRY AND CARTOGRAPHY)

OF

THE EUROPEAN ORGANIZATION FOR EXPERIMENTAL PHOTOGRAMMETRIC RESEARCH (OEEPE)

Official Publication N° 23

Wir bitten, die vorgedruckte Empfangsbestätigung auf der unteren Hälfte dieses Blattes herauszutrennen, Ihre Adresse, bei Änderung auch die frühere, einzutragen und die Karte zurückzusenden. Bei mehreren Heften in einer Lieferung genügt es, nur eine Karte zurückzusenden, wenn die Nummern der anderen Hefte auf dieser notiert sind. *Sollten die Bestätigungskarten ausbleiben, müssen wir annehmen, daß Sie am Tauschverkehr mit uns nicht mehr interressiert sind, und weitere Lieferungen einstellen.*

Please detach the acknowledgement of receipt attached below, enter your address, in case of change enter your former address, too, and return the card.
If a shipment comprises several volumes, please return one card only listing all numbers. *If the receipt is not acknowledged we understand that you are no longer interested in our exchange of publications and that you wish to be cancelled from our mailing list.*

Ayez la bonté de détacher l'accusé de réception que vous trouvez sur la partie inférieure de cette page, d'y inscrire votre adresse et, le cas échéant, votre ancienne adresse et de retourner la carte. Si la livraison se compose de plusieurs fascicules, il suffit de ne retourner qu'une seule carte portant également les numéros des autres fascicules. *Le fait d'avoir expédié un envois sans avoir reçu la carte de réception nous donnera raison à supposer que vous n'êtes plus intéressés à continuer l'échange de publications et que vous approuverez l'arrêt de nos envois.*

Le rogamos sacar el acuse de recibo figurando en la parte inferior de esta página, inscribir su dirección y si ha cambiado, inscribir la dirección anterior, y retornar esta tarjeta.
Si el envío se compone de varios fascículos es bastante retornarnos una sola tarjeta indicando también los números de los otros fascículos. *Si quedamos sin acuses de recibo tenemos que suponer que ya no tiene Vd interés en nuestro cambio de publicaciones y que desea Vd suspender otros envíos.*

Absender:
Sender:
Expéditeur:
Expedidor:

Es sind hier eingegangen:
We received:
Nous avons reçu
Han llegado en nuestras manos

OEEPE
Publ. off. No. 23

Institut für Angewandte Geodäsie
Richard-Strauss-Allee 11
D-6000 Frankfurt am Main 70
Germany · Allemagne · Alemania

June 1990

EUROPEAN ORGANIZATION FOR EXPERIMENTAL PHOTOGRAMMETRIC RESEARCH

AUTOMATIC DIGITIZING

REPORT SUBMITTED BY A WORKING GROUP
OF
COMMISSION D (PHOTOGRAMMETRY AND CARTOGRAPHY)
OF
THE EUROPEAN ORGANIZATION FOR EXPERIMENTAL
PHOTOGRAMMETRIC RESEARCH (OEEPE)

Official Publication N° 23

ISSN 0257-0505

Printed and published by the Institut für Angewandte Geodäsie, Frankfurt am Main

EUROPEAN ORGANIZATION
for
EXPERIMENTAL PHOTOGRAMMETRIC RESEARCH

STEERING COMMITTEE

(composed of Representatives of the Governments of the Member Countries)

President:	Administrateur-Général J. DE SMET Institut Géographique National 13, Abbaye de la Cambre B-1050 Bruxelles	Belgium
Members:	Dipl.-Ing. H. NOWAKOWSKI Bundesamt für Eich- und Vermessungswesen Krotenthallergasse 3 A-1080 Wien	Austria
	Ir. J. VEREECKEN Director of the Dept. of Photogrammetry Institut Géographique National 13, Abbaye de la Cambre B-1050 Bruxelles	Belgium
	Mr. O. BRANDE-LAVRIDSEN Laboratoret for Fotogrammetri og Landmåling Ålborg Universitets Center Fibigerstraede 11 DK-9220 Ålborg Ø	Denmark
	Mr. J. KRUEGER Matrikeldirektoratet Opmalingsafdelingen Titangade 13 DK-2200 Copenhagen N	
	Prof. Dr.-Ing. F. ACKERMANN Institut für Photogrammetrie der Universität Stuttgart Keplerstrasse 11 D-7000 Stuttgart 1	Federal Republic of Germany
	Abt. Dir. Dr.-Ing. H. BAUER Niedersächsisches Landesverwaltungsamt — Landesvermessung — Warmbüchenkamp 2 D-3000 Hannover 1	
	Präsident und Prof. Dr.-Ing. H. SEEGER Institut für Angewandte Geodäsie Richard-Strauss-Allee 11 D-6000 Frankfurt am Main 70	
	Mr. M. JAAKKOLA National Board of Survey Box 84 SF-00521 Helsinki 52	Finland

Prof. Dr. E. KILPELÄ — Finland
Institute of Photogrammetry Helsinki
University of Technology
SF-02150 Espoo 15

Mr. G. DUCHER — France
Directeur de la Recherche
Institut Gēographique National
2, Avenue Pasteur
F-94160 Saint Mandē

Mr. J. DESUDDE
Institut Gēographique Nation
Aērodrome de Creil
F-60107 Creil

Prof. R. GALETTO — Italy
Istituto di Topografia della
Facolta di Ingegneria della Universita
Via Strada Nuova
I-27100 Pavia

Dr. Eng. L. SURACE
Geographical Military Institute
Via Cesare Battista 8—10
I-50122 Firence

Prof. Dr. G. LIGTERINK — Netherlands
Technical University Delft
Thijsseweg 11
NL-2600 GA Delft

Ir. C. J. REMYNSE
Hoofddirecteur Hoofddirectie
v. h. Kadaster en de Openbare Registers
Waltersingel 1
NL-7300 GH Apeldoorn

Mr. Ø. BRANDSÆTER — Norway
Deputy Director
Mapping Division
Statens Kartverk
N-3500 Hønefoss

Mr. I. INDSET
Statens Kartverk
N-3500 Hønefoss

Prof. J. TALTS — Sweden
National Land Survey of Sweden
S-80112 Gävle

Prof. K. TORLEGÅRD
Royal Institute of Technology
Dept. of Photogrammetry
S-10044 Stockholm 70

Prof. Dr. O. KÖLBL — Switzerland
Institut de Photogrammētrie, EPFL
GR-Ecublens
CH-1015 Lausanne

Mr. R. KNÖPFLI
Vize-Direktor
Bundesamt für Landestopographie
Seftigenstrasse 264
CH-3084 Wabern

Major General C. ULKELUL — Turkey
Ministry of National Defence
General Command of Mapping
TR-06100 Ankara

Director General P. Mc MASTER — United Kingdom
Ordnance Survey
Romsey Road
Maybush
Southampton S09 4DH

Dr. I. J. DOWMAN
Dept. of Photogrammetry and Surveying
University College London
Gower Street 6
London WC 1E BT

SCIENCE COMMITTEE

Dr. I. J. DOWMAN — United Kingdom
Dept. of Photogrammetry and Surveying
University College London
Gower Street 6
London WC 1E BT

EXECUTIVE BUREAU

Mr. J. KURE
Secretary General of the OEEPE
International Institute for Aerospace Survey
and Earth Sciences (ITC)
350 Boulevard 1945, P. O. Box 6
NL-7500 AA Enschede (Netherlands)

Prof. Dr. H. G. JERIE
International Institute for Aerospace Survey
and Earth Sciences (ITC)
350 Boulevard 1945, P. O. Box 6
NL-7500 AA Enschede (Netherlands)

General Ing. M. CARLA
Via Giambologna 14
I-50132 Firenze (Italy)

HONORARY SECRETARY GENERAL

Ir. R. VERLAINE
Secretaire General Honoraire
36 Avenue E. Digneffe
Liège (Belgium)

SCIENTIFIC COMMISSIONS

Commission A — **Aerotriangulation**

President: Mrs. P. NOUKKA
National Board of Survey
Box 84
SF-00521 Helsinki 52

Commission B — **Digital Elevation Models**

President: Dr. K. TEMPFLI
International Institute for Aerospace Survey
and Earth Sciences (ITC)
350 Boulevard 1945, P. O. Box 6
NL-7500 AA Enschede

Commission C — **Large Scale Restitution**

President: Prof. Dr. O. KÖLBL
Institut de Photogrammētrie, EPFL
GR-Ecublens
CH-1015 Lausanne

Commission D — **Photogrammetry and Cartography**

President: Major General C. THOMPSON
Burgh House
Burgh by Sands
Carlisle CA5 6AN
United Kingdom

Commission E — **Topographic Interpretation**

President: Dr. B.-S. SCHULZ
Institut für Angewandte Geodäsie
Richard-Strauss-Allee 11
D-6000 Frankfurt am Main 70

Commission F — **Fundamental Problems of Photogrammetry**

President: Prof. Dr. G. LIGTERINK
Technical University Delft
Thijsseweg 11
NL-2600 GA Delft

APPLICATION COMMISSIONS

Commission I — **Topographic Mapping**

President: Mr. M. J. D. BRAND
Director
Ordnance Survey of N. Ireland
Colby House, Stranmillis Court
Belfast BT 9 5BJ
United Kingdom

Commission II — **Cadastral Mapping**

President: Ir. L. A. KOEN
Hoofddirectie
v. h. Kadaster en de Openbare Registers
Waltersingel 1
NL-7300 GH Apeldoorn

Commission III — **Engineering Surveys**

President: Mr. A. FLOTRON
Bahnhofstrasse
CH-3860 Meiringen

Commission IV — **Environmental/Thematic Surveys**

Commission V — **Land Information Systems**

President: Prof. Dr. W. GÖPFERT
Technische Hochschule Darmstadt
Petersenstrasse 13
D-6100 Darmstadt

DK 681.3.01/.04
771.429
657.47:65.012.12

AUTOMATIC DIGITIZING

REPORT SUBMITTED BY A WORKING GROUP
OF
COMMISSION D (PHOTOGRAMMETRY AND CARTOGRAPHY)
OF
THE EUROPEAN ORGANIZATION FOR EXPERIMENTAL
PHOTOGRAMMETRIC RESEARCH (OEEPE)

(with 85 Figures, 6 Tables and 6 Appendices)

editors:

Jane Drummond
Rob van Essen
Marijn Bosma
Pascal Boulerie

working group national representatives:

Henri Aalders	(Netherlands)
Kurt Brassel	(Switzerland)
Karin Haldrup	(Denmark)
Taipo Keisteri	(Finland)
Margaret Robb	(Norway)
John Rollin/Dave Sharman	(United Kingdom)
Francois Salge	(France)
Olivier Swartenbroekx	(Belgium)
Juri Talts	(Sweden)
Erich Wilmersdorf	(Austria)

MAY 1990

PREFACE

The OEEPE Commission D Working Group 'Automatic Digitising' was established in 1985 to follow up the Commission D 'Test of Digitising Methods' which recommended further investigation of vectorising software and the procedures used for adding 'intelligence' to automatically digitised cartographic products.

This report of the Working Group sets out the results of their investigations from 1985 to 1990. It contains a wealth of information and detail on the present status of automatic digitising, including a first evaluation of the use of low-cost scanners for cartographic digitising. It also includes an important analysis of some of the problems found in vectorising and automatic feature coding, and offers some proposals for their solution.

The OEEPE Steering Committee have decided that the results of this work should be published without delay, and I therefore have pleasure in commending this work to the widest readership. The real value of this report lies in the detail of the results and their analysis. However this must be within the overall context, objectives and main conclusions of the report.

The Objectives of the Working Group were initially set as:

1. To examine the effectiveness of various vectorising approaches.

2. To collect and disseminate information on map production processes which have automatic digitizing at the input stage.

3. To collect and disseminate information on methods to attach intelligence automatically.

Subsequently a further objective was added, namely to assess the potential of low cost scanners.

The Work of the Working Group was carried out in four parts:

1. Circulation and analysis of three questionnaires to determine the extent to which automatic digitising was being used in production agencies, the procedures being adopted and some of the vectorisation and feature coding problems commonly found.

2. Assessment of the potential of low-cost scanners (Phase I).

3. Examination of some problems in, and proposals of some solutions to vectorising Phase III).

4. Examination of some problems in, and proposals of some solutions to feature coding and adding further intelligence to vectorised data (Phase III).

The Results of the Working Group are reported as follows:

PART A The Project and the Questionnaires (pages 1-27).

PART B The potential of low cost (pages 28-77).

PART C Vectorising and feature coding (pages 78-164).

In her own Acknowledgements Jane Drummond has paid tribute to the members of the Working Group and the many other people who have contributed to their work. I would like to add my own commendation for the unstinting effort that Jane Drummond has contributed herself. Without her enthusiasm, imagination and determination to produce the results the work itself would never have advanced much beyond my first tentative suggestions, and this report would not have been possible.

Chris Thompson
President OEEPE Commission D

Carlisle, June 1990

CONTENTS

Automatic Digitizing

Report submitted by a Working Group
of
Commission D (Photogrammetry and Cartography)
of
the European Organization for Experimental
Photogrammetric Research (OEEPE)

ABSTRACT: This publication reports on the research of the Automatic Digitizing Working Group of Commission D (Photogrammetry and Cartography) of the European Organisation for Experimental Photogrammetric Research (OEEPE). The working group was established in 1985 and completed its tasks in 1990.

The remit of the working group was: to examine the effectiveness of various vectorizing approaches; to collect and disseminate information on map production processes using automatic digitizing; and to collect and disseminate information on methods used to attach intelligence automatically.

The group commenced its work by designing and circulating a series of three questionnaires to survey the status of automatic digitizing in Europe. The results of this survey are presented in detail in Part A of this report. Briefly, the results indicated that only 70 % of digital map-makers in Europe used automatic digitizing, and the working group decided this was because of high equipment costs. Of those organisations using automatic digitizing, most (87 %) required vectorized and feature coded data, but the next highest requirement (63 %) was for raster data without feature codes. On the benefit/disbenefit side 63 % found some financial and productivity advantages after introducing automatic digitizing, but 68 % of the organisations still had problems after the implementations of their automatic digitizing systems. The last questionnaire of the survey determined specifically what the outstanding problems were by asking the organisations to comment on the problems they expected when scanning and processing 17 example documents.

On completion of the survey stage the group designed its research programme. There were three phases to this. Phase I involved an investigation of the low-cost scanners now coming onto the market; this is reported in Part B. Phases II and III involved examining outstanding problems in vectorizing and automatic feature coding, and proposing and testing solutions to them; this is reported in Part C.

In Phase I we examined 46 scanners costing less than Dfl 20,000 (approx. US-$ 10,000) via the commercial and academic literature, trade fairs, symposia, etc. Details including cost, format, resolution, output data format, etc. are given in this report. Three scanners were tested for completeness of scanning, and we found completeness correlating highly with resolution. With a resolution of 63 microns all detail on our test documents (a utility map and a set of crosses) was detected, but there was some loss of detail when scanning at 127 microns. Two scanners were also tested for scanning accuracy. The first, which had only A4 format, introduced a root mean square error of 0.15 mm which was entirely comparable with the scanners which have been used in cartography for many years. The second scanner introduced a root mean square error of about 0,4 mm (unless

a large number of control points was used), but its larger format (A0 or A1) brought some benefits, and we also found that used to scan a contour separate the second scanner provided input for creating an entirely acceptable DTM.

In Phase II we examined the problems of vectorizing. We regarded vectorizing as a six step task: 1) noise removal; 2) skeletonizing; 3) node improvement; 4) line tracking; 5) segment merging; and 6) topological reconstruction. Phase II is reported in Part C of this publication.

Noise may be "salt and pepper" noise which is isolated from desired features, noise intersecting desired features, or artefacts introduced by the scanning process when map lines are close or angled sharply (this last problem is called "bridging"). By regarding noise as surrounded by a chain of short vectors beginning and ending near each other, a noise removal procedure was developed and successfully implemented. By identifying the presence of "white noise" in scanned data we were able to locate where bridging was likely to occur, and successfully prevented the formation of these artefacts.

Although **Skeletonization** is most efficiently performed in 3 x 3 matrices, this small matrix size may result in the unnecessary disconnection of features. We examined many skeletonizing algorithms, and found approaches published by *Pavlidis* first in 1980 presented the fewest problems. We tested *Pavlidis*' algorithm exhaustively, and found it generally effective, although we achieved improved performance with some modifications of our own.

Skeletonization attempts to find the medial axis of a line, but collapses at junctions — incorrectly finding the junction node point. "Node improvement" is needed — but only at junctions. We developed two techniques for identifying junctions (one based on Freeman codes and the other on vectors). Once identified the junctions can be analysed to find a good node location. We termed this "skeletonization with node improvement" and produced a new algorithm, called TON (for Thinning Of Nodes) which was successfully implemented.

Line Tracking succeeds skeletonization and node determination. Conceptually line tracking is easy, but it appeared to us that present approaches are time consuming. We therefore developed and implemented a successful and rather fast new line tracking procedure.

Segment Merging is required because the processing software may have divided up the original scan data into small patches for computational efficiency, or because the original map symbolisation (such as footpaths as pecked lines) was segmented. We addressed the second cause and successfully implemented a new algorithm for merging simple line styles.

The last of our six listed vectorization steps is **Topological Reconstruction.** Our efforts in this direction consisted mainly of a literature review, but one aspect of topological reconstruction is automatic feature coding, and this represents Phase III of the research.

All new algorithms developed in Phase II have been implemented in the DEC VAX 3600 environment and tested on fictitious data sets. More exhaustive testing on real data sets is recommended.

Automatic Feature Coding (Phase III) can be aided by component labelling, which codes regions (or polygons) by identifying and labelling each pixel belonging to a polygon, and labelling the pixels of different polygons distinctively. We successfully implemented this. Other approaches include template matching raster images of features (especially symbols), template matching using Freeman codes, using contour vectors, numerical pattern recognition, and the contextual approach presented by *De Simone* in 1986. Very full information from the Working Group member from Austria has indicated a high level of satisfaction with the automatic feature coding capability in his professional environment; in this case the scanned originals are good quality, monochromatic, large scale municipal maps, not having text and linework intersecting. Our work on Phase III is reported in Part C.

Several appendices are bound with this report, and they include copies of three survey questionnaires and details of the literature database (which was a by-product of the working group's activities).

RÉSUMÉ: La présente contribution traite des activités de recherche du groupe de travail «Numérisation Automatique» de la Commission D (Photogrammétrie et Cartographie) de l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales. Ce groupe de travail a été créé en 1985 et a achevé ses tâches en 1990.

Les thèmes du groupe de travail étaient les suivants: examen de l'efficience de diverses méthodes de vectorisation, recueil et dissémination d'informations concernant des processus de production de cartes utilisant la numérisation automatique, ainsi que recueil et diffusion d'informations sur des méthodes servant à relier l'intelligence par voie automatique.

Le groupe a commencé son travail par l'élaboration et la distribution d'une série de trois questionnaires servant à faire l'inventaire de la numérisation automatique à l'échelon européen. Les résultats de cette enquête sont détaillés dans la partie A de ce rapport. En bref, les résultats ont montré que seulement 70 % des producteurs européens de cartes sous forme numérique se servent de procédés automatiques de numérisation, le groupe de travail ayant attribué ce fait aux coûts élevés des équipements. La plus grande partie (87 %) des organisations employant la numérisation recourt à des données vectorisées et codées par objets, tandis qu'une partie importante (63 %) opte pour des données rastrées sans clés d'objets. Pour ce qui est du rapport bénéfice/perte, 63 % ont enregistré quelques avantages financiers et en production après avoir introduit la numérisation automatique, bien que 68 % des organisations interrogées aient continué à avoir des problèmes après l'implantation de leurs systèmes de numérisation automatique. Le dernier questionnaire de l'enquête portait spécifiquement sur les problèmes restant à résoudre et demandait aux organisations de commenter leur expérience et les problèmes recontrés lors du scannage et du traitement des 17 documents fournis à titre d'exemples.

Après avoir accompli cette phase d'interrogation, le groupe a dressé un programme de recherche, en trois phases. La phase I avait pour objet l'étude des scanneurs à bas prix apparaissant maintenant sur le marché, et se trouve commentée dans la partie B. Les phases II et III s'attachaient aux divers problèmes encore à résoudre dans les domaines «vectorisation» et «codage automatique d'objets», et proposaient la mise à l'essai de différentes solutions rapportées en C.

Au cours de la phase I nous avons examiné 46 scanneurs coûtant moins de Dfl. 20.000 (approx. US-$ 10.000) à l'aide de la littérature commerciale et académique, de symposia, de foires de commerce, etc. Dans ce rapport sont indiqués des détails tels que coûts, dimensions, résolution, format des données en sortie, etc. Trois scanneurs ont été testés quant à la plénitude du scannage, et nous avons constaté que cette plénitude était en forte corrélation avec la résolution. Avec une résolution de 63 microns, tous les détails des documents d'essai (une carte d'un service public et une série de croix) ont été détectés, tandis qu'il y avait une certaine perte de détails lorsqu'on effectuait le scannage à 127 microns. Deux autres scanneurs ont été testés aussi au point de vue de leur précision de scannage. Le premier, équipé seulement pour le format A4, a fourni une erreur moyenne quadratique de 0,15 mm, ce qui est tout à fait comparable aux résultats des scanneurs utilisés en cartographie depuis bien des années. Le deuxième scanneur a fourni une erreur moyenne quadratique d'environ 0,4 mm (à moins qu'on n'utilise un grand nombre de points de contrôle), mais son format plus grand (A0 ou A1) présente quelques avantages. Nous avons également constaté que dans son utilisation pour le scannage d'une planche de courbes de niveau, le deuxième scanneur produisait des données d'entrée propres à la création d'un MNT tout à fait acceptable.

Lors de la phase II nous avons examiné les problèmes de vectorisation. Nous avons considéré la vectorisation comme une tâche en six phases: 1) élimination du bruit; 2) squelettisation; 3) amélioration des noeuds; 4) suivi des lignes; 5) fusion des segments; et 6) reconstruction topologique; la phase II est expliquée dans la partie C de la présente publication.

Le **Bruit** peut être du bruit aléatoire («sel et poivre»), indépendant des objets désirés, du bruit s'entrecoupant avec des objets désirés, ou des produits artificiels introduits par le processus de scannage lorsque les lignes sont serrées ou en angles aigus (ce dernier problème est appelé «pontage»). En regardant le bruit comme une chaîne de vecteurs courts commençant et se terminant les uns près des autres, un processus d'élimination du bruit a été développé et implanté avec succès. Grâce à l'identification de l'existence de «bruit blanc» dans des données scannées, il nous a été possible de localiser les endroits où le pontage devait probablement se produire et de prévenir ainsi avec succès la formation de ces produits artificiels.

Bien que la **Squelettisation** soit réalisée de façon très efficace avec des matrices 3 x 3, la petite taille de ces matrices peut aboutir à des discontinuités inutiles d'objets. Nous avons examiné beaucoup d'algorithmes de squelettisation et avons constaté que ce sont les approches publiées pour la première fois par *Pavlidis* en 1980 qui ont posé le moins de problèmes. Nous avons testé l'algorithme de *Pavlidis* d'une manière exhaustive et, d'une facon générale, le considérons comme efficace, quoique nous ayons pu constater une amélioration de ses résultats, une fois effectuées quelques modifications.

La squelettisation vise à trouver l'axe moyen d'une ligne, mais est en défaut aux jonctions, car elle détecte de façon incorrecte le pointnoeud de la jonction. Donc, il est nécessaire de procéder à une «amélioration du noeud» — mais seulement pour les jonctions. Nous avons développé deux techniques servant à identifier les jonctions (l'une étant basée sur des codes Freeman et l'autre sur des vecteurs). Une fois identifiées, les jonctions peuvent être analysées afin de trouver une bonne position du noeud. Nous avons désigné cette méthode par «squelettisation avec amélioration de noeud» et avons produit un nouvel algorithme, appelé ADN (Amincissement Des Noeuds), qui a pu être mis en oeuvre sans problèmes.

Le **Suivi des Lignes** suit la squelettisation et la détermination des noeuds. Du point de vue conceptuel, le suivi des lignes est facile, bien que nous ayons l'impression que les approches actuelles exigent beaucoup de temps. C'est pourquoi nous avons développé et établi un nouveau procédé de poursuite des lignes, efficace et assez rapide.

La **Fusion de Segments** est nécessaire du fait que le logiciel de traitement peut diviser les données originales de scannage en petites zones pour des raisons d'efficacité de calcul, ou parce que la symbolisation de la carte originale e été segmentée (p.ex. sentiers sous forme de lignes en tireté). Nous avons abordé la deuxième cause et avons établi un nouvel algorithme pour la fusion des tirets séparés d'une même ligne.

La dernière de nos six phases de vectorisation est la **Reconstruction Topologique**. Nos efforts à cet égard ont principalement consisté en une revue de la littérature spécialisée en mettant l'accent sur l'aspect du codage automatique d'objets, ce qui représente la Phase III de notre recherche.

Tous les nouveaux algorithmes développés en Phase II ont été implantés dans l'environnement DEC VAX 3600 et testés sur des séries de données fictives. Des tests plus étendus sur des séries de données réelles sont néanmoins recommandés.

Le **Codage Automatique d'Objets** (Phase III) peut être facilité lorsqu'on attribue des attributs aux composants, grâce auxquels sont codées des régions (ou polygones) en identifiant chaque pixel appartenant à un polygone et en lui apposant un label, tout en se servant de labels distinctifs pour le marquage de polygones différents. Il nous a été possible de réaliser cela sans difficulté. D'autres procédés prévoient la «mise en correspondance avec des gabarits issus des images scannées d'objets» (particulièrement des symboles). Ce procédé se sert de codes Freeman en utilisant soit des vecteurs de contour, soit la reconnaissance numérique d'objets, ainsi que la méthode contextuelle présentée par *De Simone* en 1986. Des informations très complètes de la part du représentant autrichien au groupe de travail ont mis en évidence le haut degré de satisfaction qu'il a pu atteindre dans son environnement professionnel grâce aux possibilités actuelles en matière de codage automatique d'objets. Dans ce dernier cas les originaux scannés étaient d'une bonne qualité, monochromes, pris dans des cartes municipales à grande échelle, sans comporter d'intersections entre texte et lignes. Nos activités relatives à la Phase III sont décrites dans la partie C.

Le rapport contient plusieurs annexes, ainsi que les copies des trois questionnaires et des détails relatifs à la base de données bibliographique (un sous-produit des activités du groupe de travail).

ZUSAMMENFASSUNG: In dieser Publikation wird über die Forschungsergebnisse der Arbeitsgruppe „Automatisches Digitalisieren" der Kommission D (Photogrammetrie und Kartographie) der OEEPE berichtet. Die Arbeitsgruppe wurde im Jahr 1985 eingerichtet, und im Jahr 1990 war die ihr gestellte Aufgabe erfüllt.

Der Arbeitsgruppe war aufgetragen worden, die Wirksamkeit verschiedener Vektorisierungsmethoden zu untersuchen, Informationen über Kartenproduktionsprozesse zu sammeln und zu verbreiten, bei denen automatische Digitalisierung angewandt wird und Informationen über Methoden mit automatischer Nutzung von Intelligenz zu sammeln und zu verbreiten.

Die Gruppe begann ihre Arbeit mit dem Entwurf und dem Versand einer Serie von drei Fragebögen, um den Stand der automatischen Digitalisierung in Europa zu bestimmen. Die Ergebnisse dieser Umfrage werden detailliert im Teil A dieses Berichts vorgestellt. Kurz gesagt, aus den Ergebnissen geht hervor, daß nur 70 % derjenigen, die in Europa digitale Karten erzeugen, automatisches Digitalisieren benutzen; die Arbeitsgruppe vermutete, daß dies wegen der hohen Gerätekosten so ist. Die meisten (87 %) der Organisationen, welche automatisches Digitalisieren anwenden, benutzen objectcodierte Vektordaten; dieser Zahl am nächsten (mit 63 %) kommt die Nutzung von Rasterdaten ohne Objektkode. Auf der Kosten/Nutzen-Seite sehen 63 % finanzielle Vorteile und Steigerung der Produktivität nach Einführung der automatischen Digitalisierung, jedoch 68 % der Organisation hatten nach der Implementierung ihres automatischen Digitalisiersystems immer noch Probleme. Der letzte Fragebogen der Umfrage befaßte sich speziell mit diesen Problemen, wobei die Organisationen gebeten wurden, sich zu den erwarteten Problemen zu äußern, die sie beim Scannen und Bearbeiten von 17 Beispielen erwarten.

Nach Abschluß der Befragungsaktion stellte die Gruppe ihr Forschungsprogramm auf, wofür drei Phasen vorgesehen waren. Phase I betraf eine Untersuchung über „Billig-Scanner", die zu dieser Zeit am Markt waren; darüber wird im Teil B berichtet. In den Phasen II und III wurden anstehende Probleme der Vektorisierung und der automatischen Objekt-Kodierung behandelt und Vorschläge zur Lösung dieser Probleme getestet; darüber wird im Teil C berichtet.

In die Phase I schlossen wir 46 Scanner ein, die weniger als 20 000 Dfl (etwa 10 000 US-$) kosteten, über die wir Informationen aus kommerziellem und wissenschaftlichem Schrifttum, aus Handelsmessen und Symposien usw. bezogen. Details einschließlich Preise, Format, Auflösung, Output-Datenformat usw. werden in dem Bericht erwähnt. Drei Scanner wurden geprüft auf die Vollständigkeit der Abtastung, wobei festgestellt wurde, daß die Vollständigkeit mit der Auflösung stark korreliert ist. Bei einer Auflösung von 63 µm wurden alle Details unseres Testbeispiels (eine thematische Karte und ein Satz Kreuze) wiedergegeben, bei einer Auflösung von 127 µm trat ein Detailverlust auf.

Zwei Scanner wurden auch auf Abtastgenauigkeit getestet. Der erste, der nur A4-Format besaß, erbrachte einen mittleren Fehler von 0,15 mm, was durchaus mit den Scannern vergleichbar war, die in der Kartographie über viele Jahre gebraucht wurden. Der zweite Scanner erbrachte einen mittleren Fehler von 0,4 mm (sofern nicht eine große Anzahl von Kontrollpunkten benutzt wurden), aber sein größeres Format (A0 oder A1) brachte einige Vorteile, und bei der Abtastung einer Höhenlinienplatte lieferte der zweite Scanner die Daten für ein durchaus brauchbares DGM.

In Phase II verfolgten wir die Probleme der Vektorbildung. Wir betrachteten die Vektorisierung als eine 6-stufige Aufgabe: 1. Beseitigung des Rauschens; 2. Skelettierung; 3. Knotenverbesserung; 4. Linienverfolgung; 5. Segmentvereinigung und 6. topologische Rekonstruktion. Über Phase II wird in Teil C dieser Veröffentlichung berichtet.

Rauschen kann „Salz- und Pfeffer"-Rauschen sein, das isoliert von den gewünschten Objekten vorkommt; Rauschen kann die gewünschten Objekte überlagern, oder es kann künstlich durch den Abtastvorgang eingeführt worden sein, wenn Kartenlinien eng geschart sind oder sich unter flachem Winkel schneiden (dieses letztere Problem kann als „Überbrücken" bezeichnet werden). Mit der Betrachtung des Rauschens als

eine Folge von kurzen Vektoren, die nahe beieinander beginnen und enden, wurde ein Verfahren zur Beseitigung des Rauschens entwickelt und erfolgreich implementiert. Erkennt man die Anwesenheit von „weißem Rauschen" in den abgetasteten Daten, so ist es möglich, die Stellen zu finden, an denen das „Überbrücken" wahrscheinlich auftreten kann. Mit dieser Erkenntnis konnten wir das Auftreten solcher Erscheinungen verhindern.

Obwohl die **Skelettierung** am besten mit 3 x 3 Matrizen durchgeführt wird, kann es bei diesen kleinen Matrizen zu Unterbrechungen der behandelten Objekte kommen. Wir untersuchten viele Skelettierungs-Algorithmen und fanden, daß die von *Pavlidis* im Jahr 1980 veröffentlichten Vorschläge am wenigsten Probleme boten. Wir testeten *Pavlidis'* Algorithmus gründlich und fanden ihn grundsätzlich richtig, dennoch konnten wir durch einige eigene Modifikationen die Ergebnisse noch verbessern. Beim Skelettieren wird versucht, die Mittelachse einer Linie zu finden; das scheiterte an Schnittpunkten — die Knotenpunkte an Linienschnitten werden verfälscht. Deshalb wird eine **Knotenverbesserung** gebraucht — jedoch nur bei Linienschnitten. Wir entwickelten zwei Techniken zur Identifizierung von Linienschnitten (eine basiert auf „Freeman Codes" und die andere auf Vektoren). Nach der Identifizierung der Schnittpunkte kann die Lage der Knoten neu festgelegt werden. Wir bezeichneten diesen Vorgang als „Skelettierung mit Knotenverbesserung" und schufen einen neuen Algorithmus, TON genannt (für „Thinning of Nodes", oder Ausdünnung von Knoten), welcher erfolgreich implementiert wurde.

Die **Linienverfolgung** schließt sich an das Skelettieren und die Knotenbestimmung an. Das Konzept der Linienverfolgung ist leicht faßbar; aber es schien uns, daß die gebräuchlichen Lösungen zu zeitaufwendig sind. Deshalb entwickelten und implementierten wir ein erfolgreiches und ziemlich schnelles neues Verfahren.

Die **Segmentvereinigung** ist notwendig, da die Verarbeitungssoftware die originären Abtastdaten aus Gründen der Rechenökonomie in kleine Abschnitte aufgeteilt haben kann oder weil die Original-Kartensymbole (wie z. B. Fußwege als unterbrochene Linien) segmentiert waren. Wir befaßten uns mit dem zweiten der erwähnten Gründe und implementierten erfolgreich einen neuen Algorithmus zur Vereinigung einfacher Liniendarstellungen.

Der letzte unserer sechs erwähnten Vektorisierungsschritte ist die **Topologische Rekonstruktion.** Unsere Bemühungen in dieser Richtung bestanden hauptsächlich in Literaturstudien; aber ein Aspekt der topologischen Rekonstruktion ist die automatische Objektkodierung, und die wird in Phase III der Untersuchung behandelt.

Alle neuen Algorithmen der Phase II wurden in eine DEC VAX 3600 Umgebung implementiert und mit fiktiven Daten getestet. Ausgiebigere Tests mit echten Datensätzen werden empfohlen.

Die **Automatische Objektkodierung** (Phase III) kann unterstützt werden mit Komponentenlabeln, mit Kode-Regionen (oder Polygonen) durch Identifizierung und Labeln eines jeden zu einem Polygon gehörenden Pixels für verschiedene unterschiedliche Polygone. Wir implementierten dies erfolgreich. Daneben existieren andere Lösungswege wie „Template matching" bei gerasterten Objekten (spezielle Symbole), „Template matching" mittels Freeman-Kodes, mittels Umrißvektoren, numerische Mustererkennung und der damit zusammenhängende Vorschlag von *De Simone* aus dem Jahr 1986. Ausführliche Informationen des österreichischen Mitglieds der Arbeitsgruppe zeigten einen hohen Stand der automatischen Objektkodierung in seiner beruflichen Praxis; hier waren die abgetasteten Originale städtische großmaßstäbige Karten von guter Qualität, monochrom und ohne Text und Gitterlinien. Über unsere Arbeit der Phase III wird im Teil C berichtet.

Zu diesem Bericht gehören verschiedene Anhänge; dazu gehören Abdrucke der drei Fragebögen und Details zu der Literatursammlung (die als Beiwerk der Aktivitäten der Arbeitsgruppe betrachtet werden kann).

ACKNOWLEDGEMENTS

This publication reports on an OEEPE study into automatic digitizing. We hope that its publication now will encourage people to examine automatic digitizers themselves, as we feel they do have a place in building Geographic Information Systems (GIS's). We have done some of the groundwork in this study and hope it will be helpful to others continuing in this field.

Very many people have kindly and willingly cooperated in this work, from the Dutch commercial and mapping world (Mr. H. Nordhoff and Ms. E. Dirkse of Geveke Electronics; Mr. B. Bruyn van Rozenburg and Mr. B. Spin of AGFA Gevaert; Mr. R.C. Schonk, Ms. G. Horselenberg, and Mr. J. Kip of Canon; Mr. L. Weldink of Nutsbedrijf IJsselmij, Enschede; and members of Publishing Technologies Nederland); from the wider European arena (Ms. P. Atkins, SysScan, U.K.; Mr. A. Van Houtte, Houston Instrument, Belgium; Dr. M. Buchroithner, Dr. W. Polzleitner, Ms. C. Fermuller, Mr. D. Strobel, DIBAG, Graz, Austria); from ITC (Mr. Y. Chung, Mr. M. Ellis, Mr. R. Haakmeester, Mr. E. Mol, Mr. L. Raidt, Mr. P. Schokker, Mr. G. Schuurman, Mr. S. van der Steen, Dr. P. Stefanovic, Ms. H. Theus, Mr. R. Vogel, and Ms. E. Weijer) and of course the working group members and the questionnaire respondents who are again mentioned in section A3. Without their help it would not have been possible to produce this report.

Turning to finance, this report represents the work of an international Working Group, and the employers of the working group members have made a significant financial contribution by allowing their employees to function as members. The International Institute of Aerospace Survey and Earth Sciences (ITC) employed Marijn Bosma as a Research Assistant for three months at the beginning of the project; this contribution was significant in 'getting things going'. IGN (France) arranged for their trainee engineer Pascal Boulerie to work on Phases II and III of this project for six months. Finally and very generously, the Steering Committee of the OEEPE, to whom this report is submitted, provided funding for a visit by Pascal Boulerie and Rob van Essen to Austria, and to employ Marijn Bosma and Rob van Essen for a further eight months in total. For these different forms of financial support we are most grateful.

Jane Drummond
Rob van Essen
Marijn Bosma
Pascal Boulerie

Enschede, May 1990

A. PROJECT DESCRIPTION AND ORGANISATION

In 1985 it was proposed at a meeting of the OEEPE (European Organisation for Experimetal Photogrammetric Research) Steering Committee, that the work of the OEEPE Working Group on a Test of Digitizing Methods [THOMPSON,1984] be continued into two areas highlighted in that project's report - namely the evolution of vectorizing software and procedures used for adding 'intelligence' to automatically digitized cartographic objects. To this end the Commission D Working Group on Automatic Digitizing was established in the Autumn of 1985.

A1. OBJECTIVES

After an initial meeting at Auto-Cart London in September 1986, and subsequent discussions with members of Commission D (the OEEPE Scientific Commission in Photogrammetry and Cartography) and Commission I (the OEEPE Application Commission in Topographic Mapping) the remit of the Working Group was established as being:

1. to examine the effectiveness of various vectorizing approaches;

2. to collect and disseminate information on map production processes which have automatic digitizing at the input stage; and,

3. to collect and disseminate information on methods used to attach intelligence automatically.

A2. STRUCTURE OF THE PROJECT

Automatic Cartographic digitizing was dominated until 1986 by extremely precise, large format, and somewhat costly scanners. remaining problems (such as in the areas of vectorizing and adding intelligence) were addressed only by a small group of map producers, the group being limited to those who could afford to buy automatic digitizing equipment and, of course, the equipment manufacturers. The task of this working group would have been relatively simple if the situation had remained thus. It could have been that a rather small group of users would have indicated, through questionnaires, where there vectorizing problems were, how they used scanners in map production, how the problems of attaching intelligence were handled, and the working group would have analysed responses and identified the more successful approaches. Additionally a researcher could have identified the different vectorizing algorithms in use, highlighted those which were more successful, and tested alternatives. The situation has now changed, for two reasons:

1. microcomputers have become widespread in the Graphics Industry;

and,

2. although vectorizing software is constantly being improved and more is becoming available, specific information about algorithms is confidential.

Because microcomputers have become widespread, even in the smallest companies in the Graphics Industry, microbased desktop publishing software incorporating low-cost scanners and raster editing has also become widespread. Cartographers using maps as sources of illustrations rather than sources of metric information may have been the first mapmakers to become aware of the potential, to mapping, of low-cost scanners. In this case we decided low-cost meant costing up to Dfl 20,000.00 (i.e. about US$10,000.00).

Another large group of microcomputer users, whose requirements overlap those of mapmakers, include Civil Engineers. Drawing programs, such as AutoCad are widely used for creating, storing, and plotting dimensioned engineering drawings. For plotting these drawings, medium resolution large format plotters (e.g. from Calcomp or Houston Instrument) are adequate; data has to be in vector form - just to drive the plotting pen, but need not be vectorized to cartographic objects (also called terrain objects, or cartographic segments, etc.), or have added intelligence, beyond only (perhaps) a layer number.

A final group of microcomputer users include Municipal Engineers. They often have vast archives of large format, large scale, utilities maps which need regular revision, and from which copies are frequently made. Until recently more forward looking members of this group met their archiving needs by microfilming, and their updating needs by traditional manual cartography. Now however they are recognizing the advantage of raster files (obtained by scanning existing maps using low cost scanners) as archives, and raster editing for updating.

In the foregoing three large professional groups have been identified. Their needs have resulted in a competitive market for low cost scanners, but their vectorizing needs are not stringent, they have hardly any requirement for adding intelligence to captured graphic data, and perhaps most importantly their positional accuracy needs are unclear.

If it is assumed that the interests of the OEEPE Commission D Working Group on Automatic Digitizing lie in precise scanning, vectorizing to terrain objects, and adding several items of intelligence to these vectorized objects then the recent ferment of activity in the sphere of microcomputer based desk top publishing, scanning, and pen plotting may offer little. But, on the other hand, the very competitive natures of the Graphics and Civil Engineering Industries (in particular) may mean that some quite low-cost scanners may actually produce quite high quality raster files, some microbased vectorizing software may vectorize to terrain objects, and some microbased editing software provide easy feature-coding and further intelligence-adding capabilities.

The task of the Working Group, which is now being completed, in the late 1980's and in an atmosphere dominated by microcomputers, must be rather different from that envisaged when the Working Group was established in the mid 1980's. The high quality automatic digitizing approaches which are well established in mapmaking (Intergraph, Lasercan, Scitex, SysScan,etc.) can be regarded as equals at one end of a spectrum whereas at the other end of the spectrum are ranked an increasing number of lower quality alternatives. The Working Group has concluded that, because it may be these lower quality approaches that break the digitizing 'logjam', the lower quality solutions must be examined as well as the established solutions found in some national Topographic Organisations. To this end the Working Group concluded that its work must be carried out in three phases:

PHASE I - Assessment of the potential of low-cost scanning;

PHASE II - Examination of some problems in and proposal of some solutions to vectorizing; and,

PHASE III - Examination of problems in and proposal of some solutions to feature-coding and adding further intelligence to vectorized data.

An important aspect of the Working Group's approach was to use questionnaires to find out what was going on in Europe with regard to computerized mapping and automatic digitizing. These are described in the section A4.

A3. WORKING GROUP MEMBERS AND COOPERATING ORGANISATIONS

The Working Group which was established consists of:

National Representatives:

MR. Erich Wilmersdorf (Austria)
MR. Olivier Swartenbroekx (Belgium)
MS. Karin Haldrup (Denmark)
MR. Francois Salge (France)
MR. Taipo Keisteri (Finland)
MR. Henri Aalders (Netherlands)
DR. Margaret Robb (Norway)
DR. Juri Talts (Sweden)
DR. Kurt Brassel (Switzerland)
MR. John Rollin, replaced by MR. David Sharman (UK)

Chair:
MS. Jane Drummond (Netherlands)

Research Assistants:
MR. H. Marijn Bosma (Netherlands)
MR. Pascal Boulerie (France)
MR. Rob van Essen (Netherlands)

These people, i.e. the national representatives, the chair, and the research assistants, are responsible for the preparation of the questionnaires, their analysis, and the production of a final report. Readers may be interested in the specific breakdown of tasks. The Working Group, at meetings held in Enschede, designed questionnaires 1 and 2 and analysed the responses. Based on this analysis Mr. Bosma designed questionnaire 3. The responses to questionnaire 3 provided the framework for the research carried out in Phases II and III. The research linked to Phases II and III was carried out by Mr. Boulerie and Mr. Van Essen. The research for Phase I was carried out by Mr. Bosma, Mr. Chung (a photogrammetry research student from S.Korea) and Ms. Drummond, with some programming support from an ITC colleague, Mr. L. Raidt. It is with great pleasure that we report that all the above mentioned Working Group members made a real contribution to our efforts; contributions involved attending meetings in Enschede, ensuring questionnaires were answered, and supplying the workers in Enschede with a stream of useful scientific papers and encouraging phone calls.

The organisations responding to the questionnaires were:

NATIONAL MAPPING ORGANISATIONS:
- Institut Geographique National, Belgium
- Geodetisk Institute, Denmark
- Matrikeldirektoratet, Denmark
- Bureau de Recherches Geologiques et Minieres, France
- Institut Geographique National, France
- Section Geographique Militaire, France
- Norsk Polar Institutt, Norway
- Norwegian Mapping Authority, Norway
- Norwegian Hydrographic Service, Norway
- National Land Survey, Sweden
- Federal Office of Topography, Switzerland
- British Geological Survey, U.K.
- Directorate of Military Survey, U.K.
- Ordnance Survey of Great Britain, U.K.
- Ordnance Survey of Northern Ireland, U.K.

COMMERCIAL MAP/ATLAS PUBLISHERS
- Total CFP, France
- Kummerley and Frey, Switzerland
- Automobile Association, U.K.
- John Bartholomew and Son, U.K.

UTILITIES: NATIONAL/REGIONAL
- Jydsk Telefon, Denmark
- PTT, Switzerland
- Yorkshire Electricity, U.K.

MUNICIPALITIES
- Magistratsdirektion der Stadt Wien, Austria
- Vermessungsamt Basel-Stadt, Switzerland
- Vermessungsamt der Stadt Zurich, Switzerland

HARDWARE MANUFACTURERS WITH ASSOCIATED COMMERCIAL DATA/SOFTWARE BUREAUX
- Laser-Scan, U.K.
- SysScan, U.K.

INDEPENDENT COMMERCIAL DATA/SOFTWARE BUREAUX
- B.Hojlund Rasmussen, Denmark
- Viak, Norway
- VBB Scan Graph, Sweden
- Viak AB, Sweden
- CadCapture, U.K
- Colourmap Scan, U.K
- Geografix, U.K
- Hilldata, U.K

ACADEMIC/RESEARCH INSTITUTES
- Institut fur Kartographie,ETH, Zurich, Switzerland

and as this report would be meaningless without their support, we are extremely grateful for their cooperation.

A4. THE QUESTIONNAIRES

Three questionnaires have been established by the Working Group.

A4.1 THE FIRST QUESTIONNAIRE

The first questionnaire (see Appendix A for a sample) was to establish the extent of digital processes in the European mapping industry, the extent of both automatic and manual digitizing, and the extent of feature-coding and further intelligence-adding. The FIGURE 1 (a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p) gives a graphic summary of the responses - but it can be seen that all respondents used digital methods, 89% in map production. 70% of the respondents used automatic digitizing, but 91% used manual digitizing. 92% of the respondents vectorized and feature coded their data. When it is remembered that Thompson [THOMPSON,1984] in 1984, found data capture (including feature-coding) using automatic digitizing methods to be from 2 to 7 times as fast as manual digitizing it is worth noting that only 70% of our respondents use automatic digitizing methods. Furthermore the use of automatic digitizers is dominated by the fifteen national mapping organisations in the survey - among smaller (less wealthy?) organisations use of automatic scanners drops to 55%. Accepting the time efficiency of automatic digitizing, it can only be assumed that market resistance results from high equipment costs. This further justifies our examination of low-cost scanners.

FIGURE 1(a-p). RESPONSES TO THE FIRST QUESTIONNAIRE

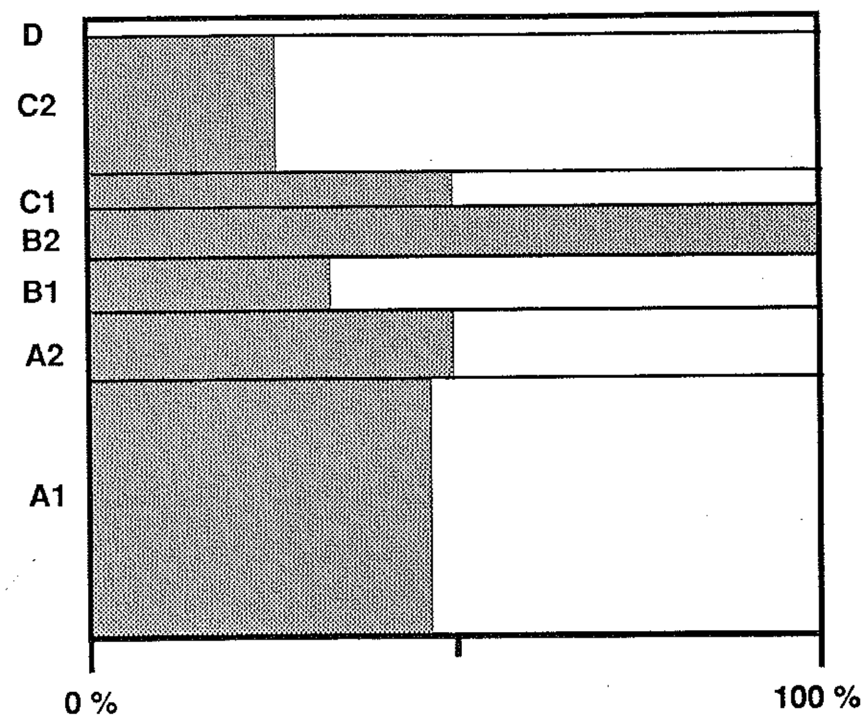
There are sixteen graphic depictions of these responses - summarised by the type of organisation as follows:

A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

The sixteen diagrams (FIGURES 1a to 1o) are in the following pages.

FIGURE 1a (responses to the First Questionnaire)

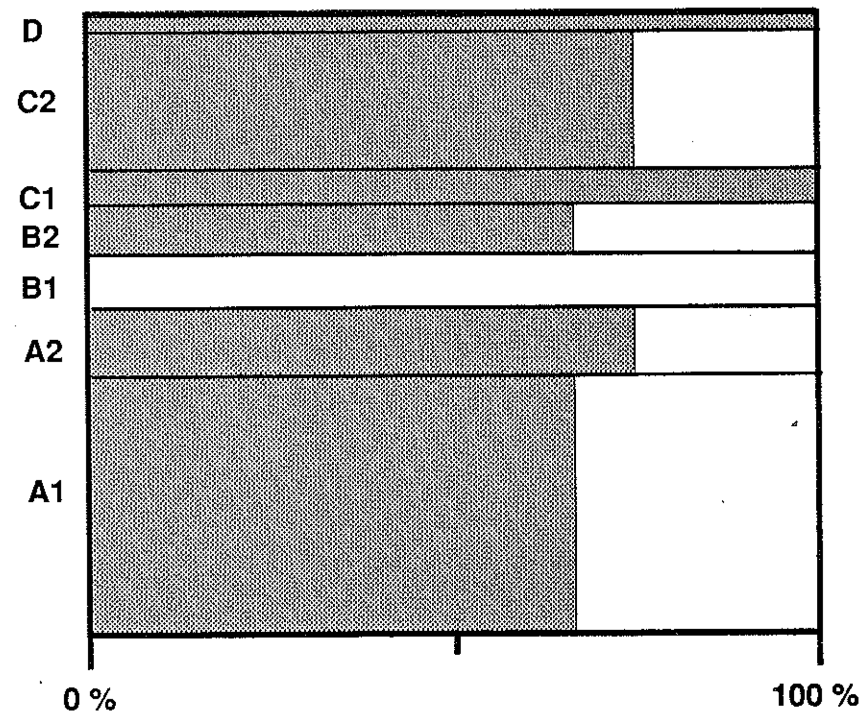
referring to: QUESTION 4.a:
Do you use computer assisted techniques in your map production at the stage of
Document Management/Map Cataloguing?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1b (responses to the First Questionnaire)

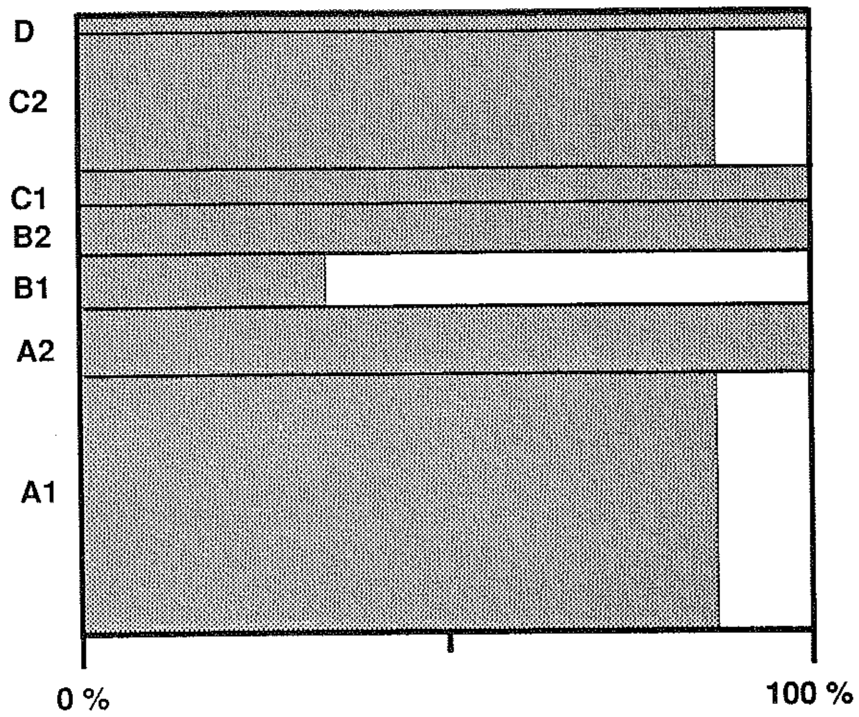
referring to: QUESTION 4.b:
Do you use computer assisted techniques in your map production at the stage of
Map Design?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1c (responses to the First Questionnaire)

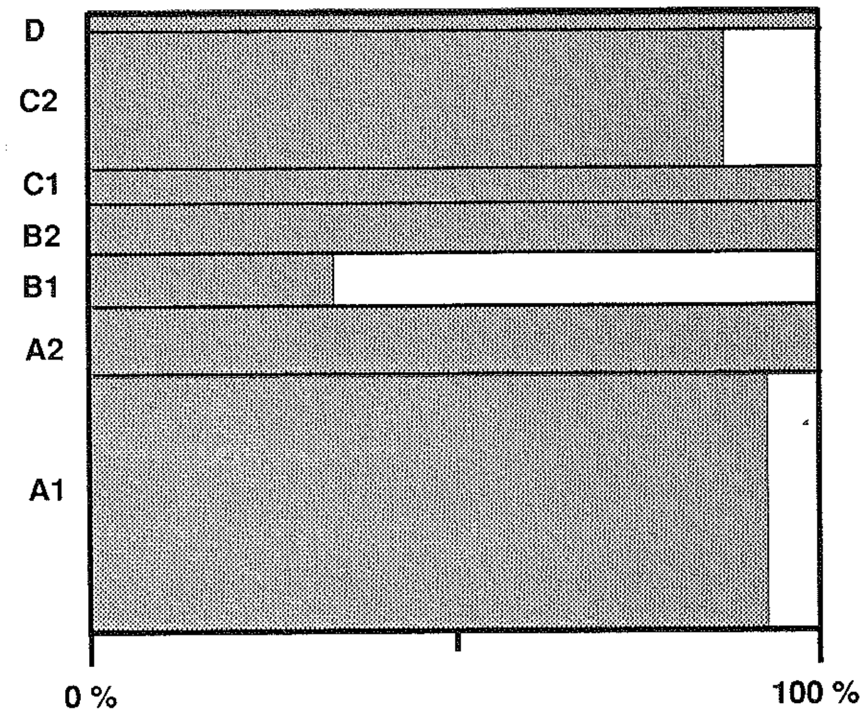
referring to: QUESTION 4.c:
Do you use computer assisted techniques in your map production at the stage of Map Data Capture?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1d (responses to the First Questionnaire)

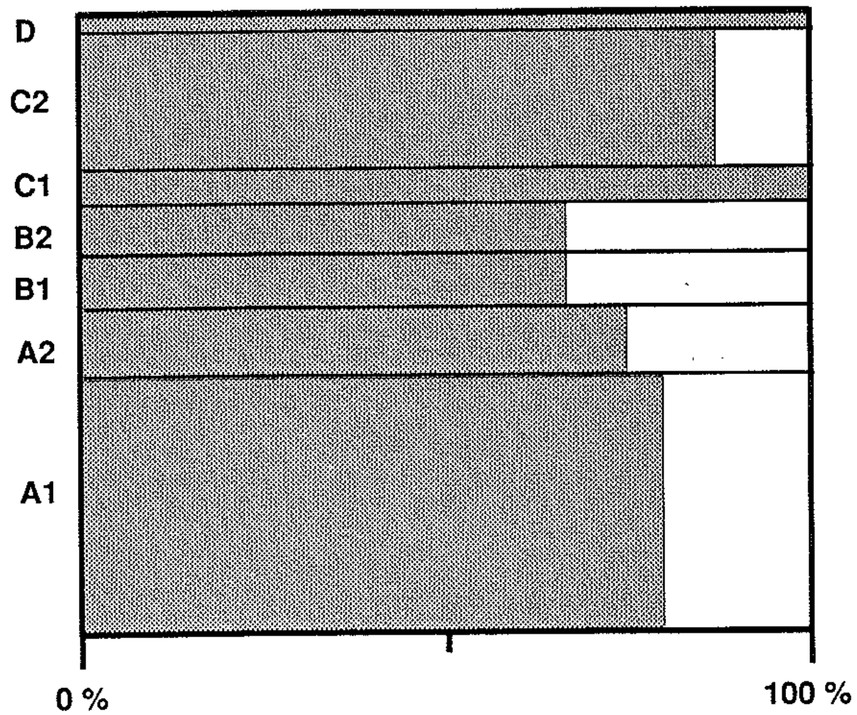
referring to: QUESTION 4.d:
Do you use computer assisted techniques in your map production at the stage of Map Data Processing?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1e (responses to the First Questionnaire)

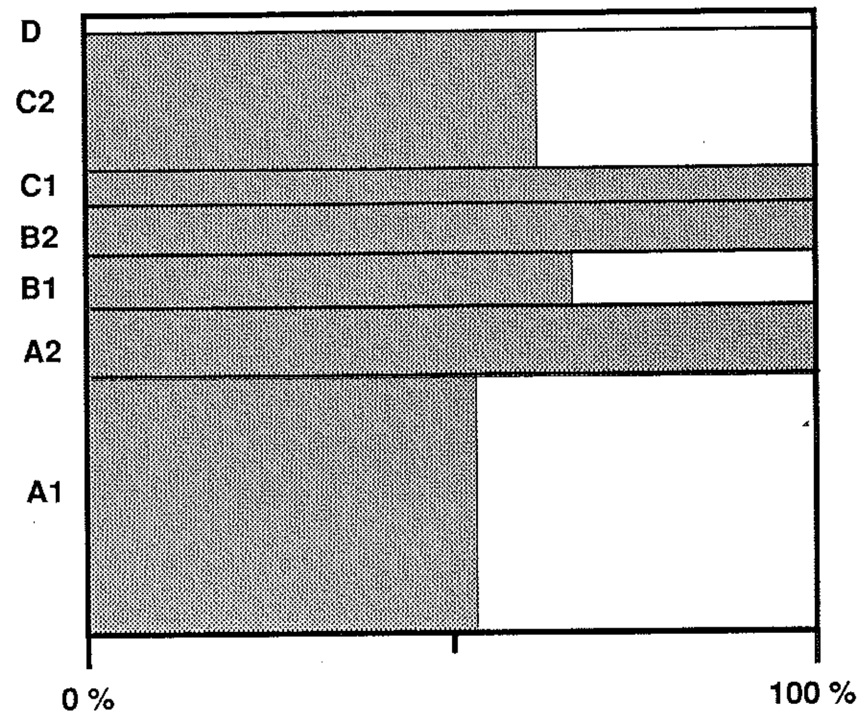
referring to: QUESTION 4.e:
Do you use computer assisted techniques in your map production at the stage of Map Presentation?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1f (responses to the First Questionnaire)

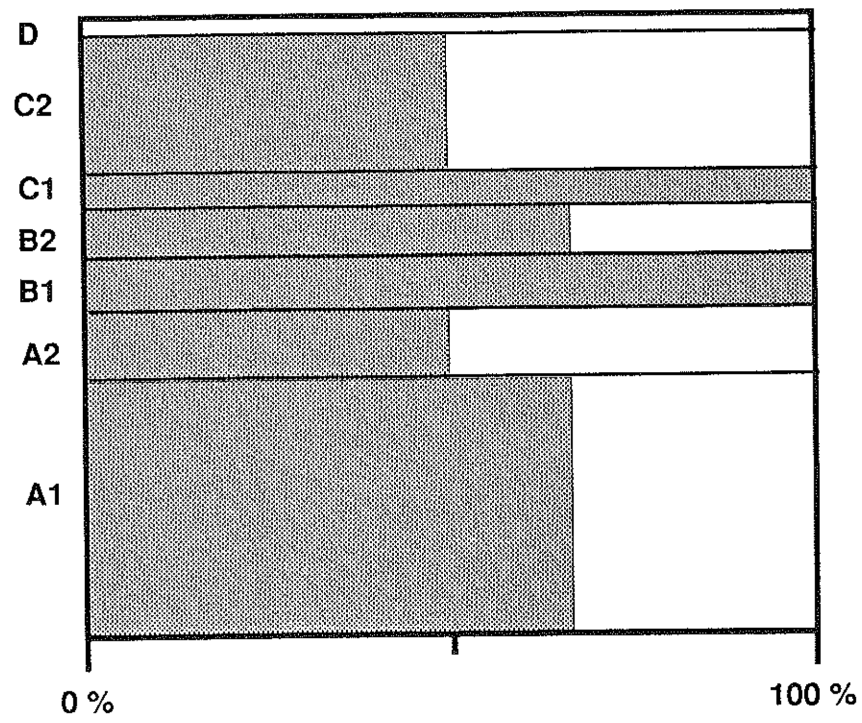
referring to: QUESTION 4.f:
Do you use computer assisted techniques in your map production at the stage of Map Archiving?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1g (responses to the First Questionnaire)

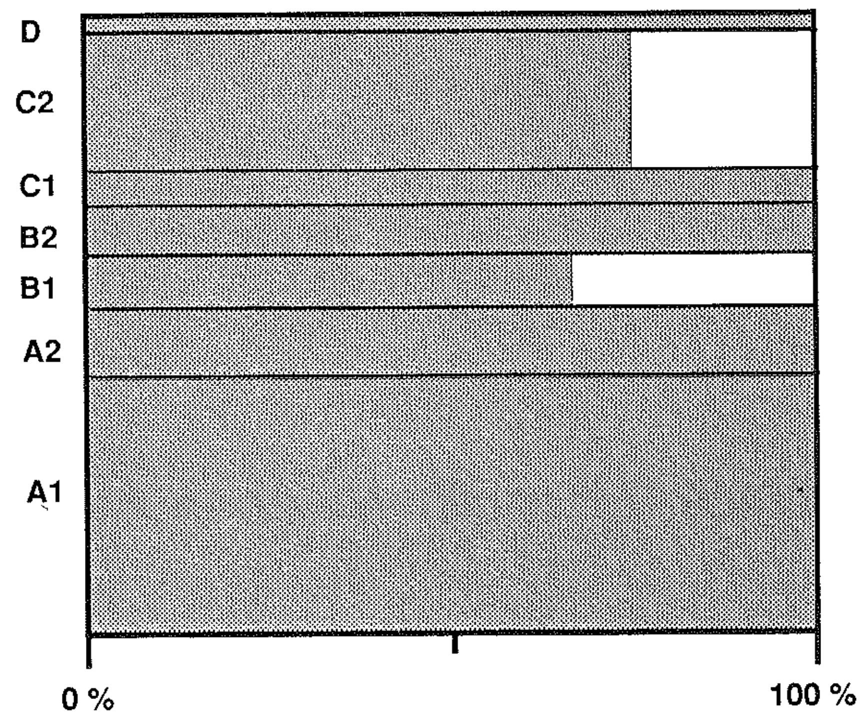
referring to: QUESTION 4.g:
Do you use computer assisted techniques in your map production at the stage of Spatial Information System Management?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1h (responses to the First Questionnaire)

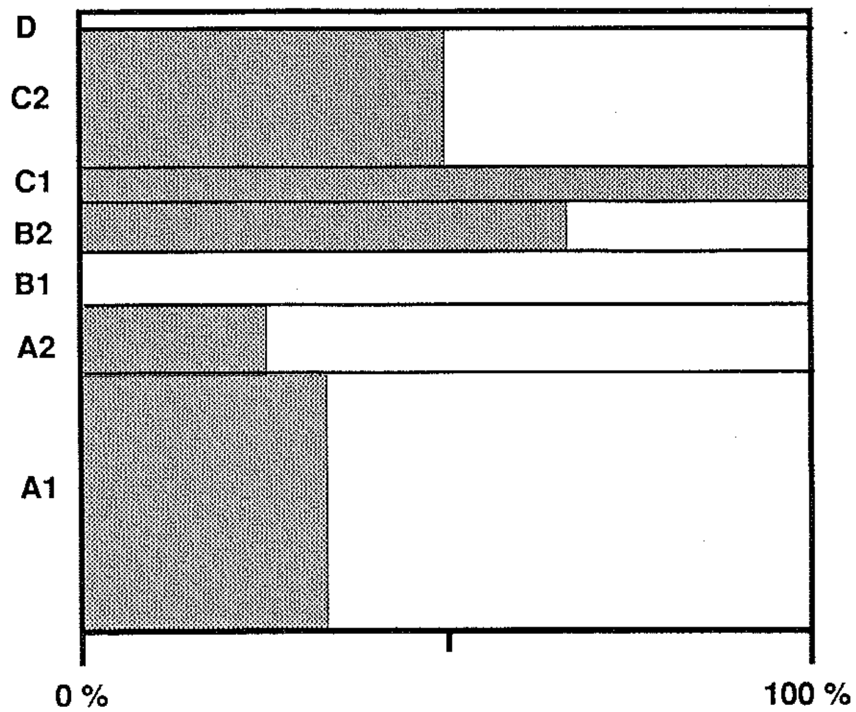
referring to: QUESTION 5.a:
Do you use manual digitizing?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1i (responses to the First Questionnaire)

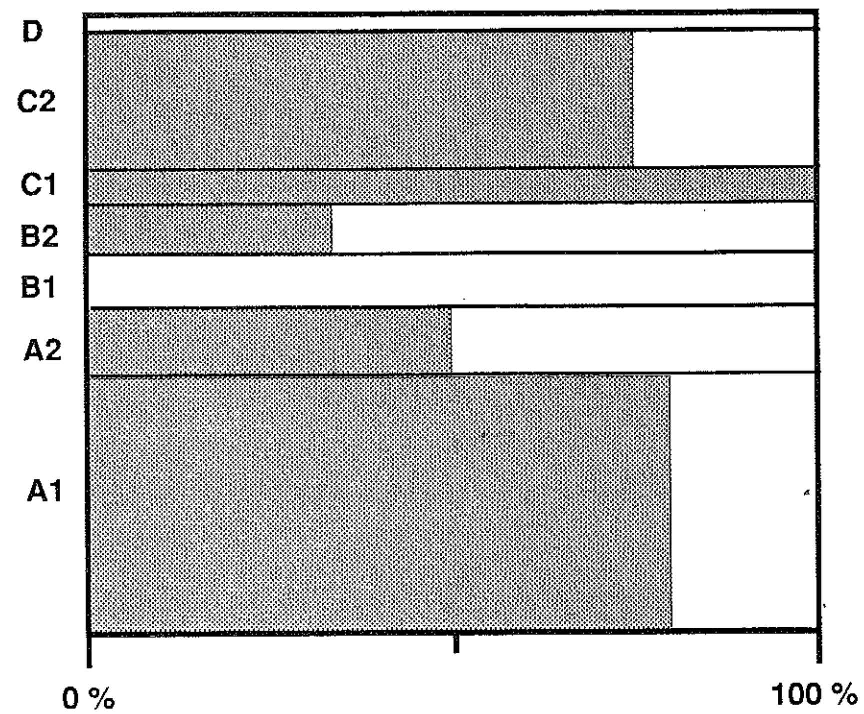
referring to: QUESTION 5.b:
Do you use semi-automatic digitizing?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1j (responses to the First Questionnaire)

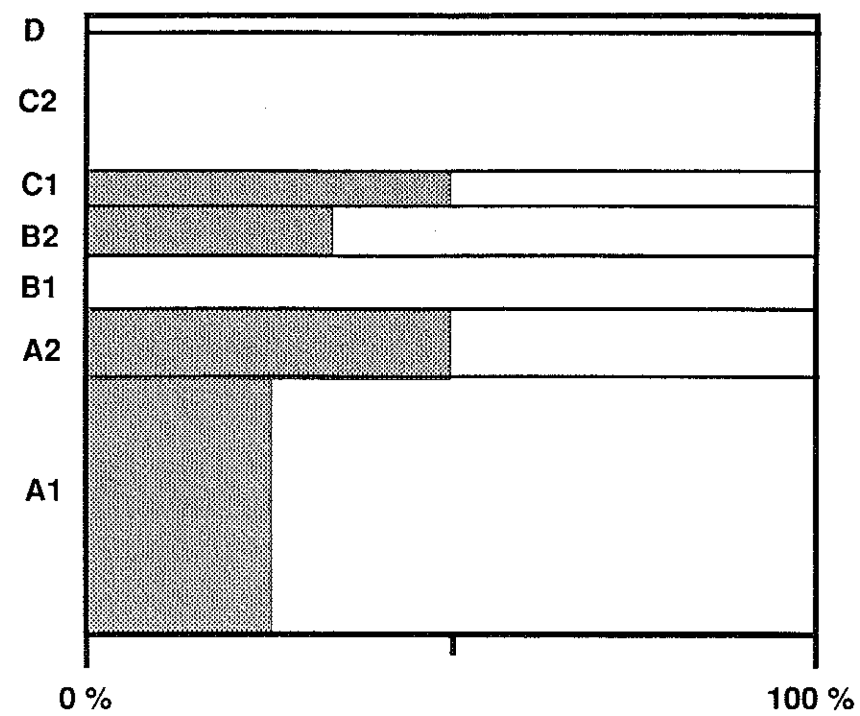
referring to: QUESTION 5.c:
Do you use automatic digitizing?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1k (responses to the First Questionnaire)

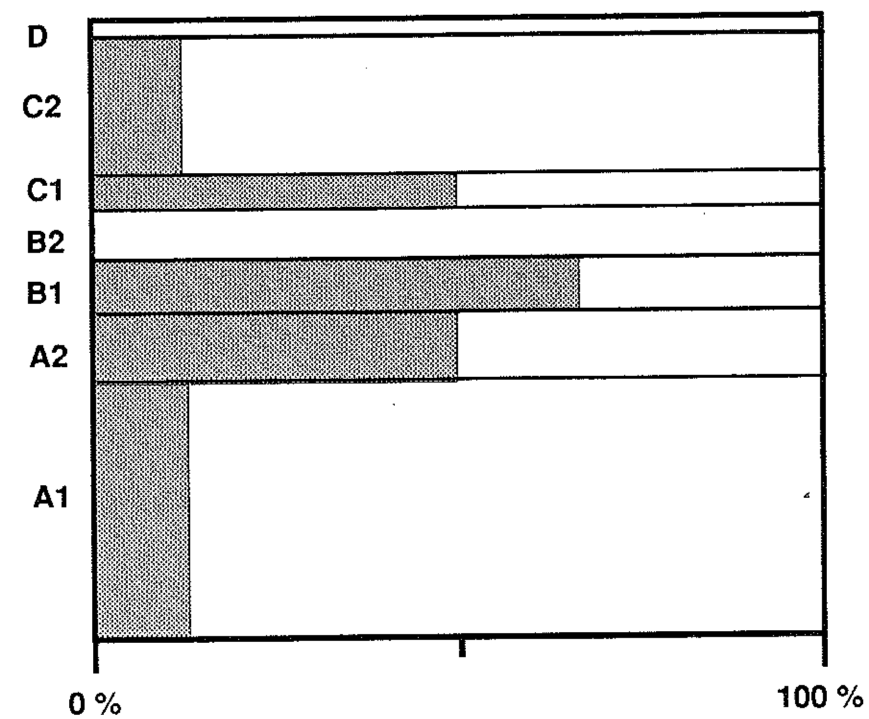
referring to: QUESTION 5.d:
Do you use purchased automatic digitized raster data?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1l (responses to the First Questionnaire)

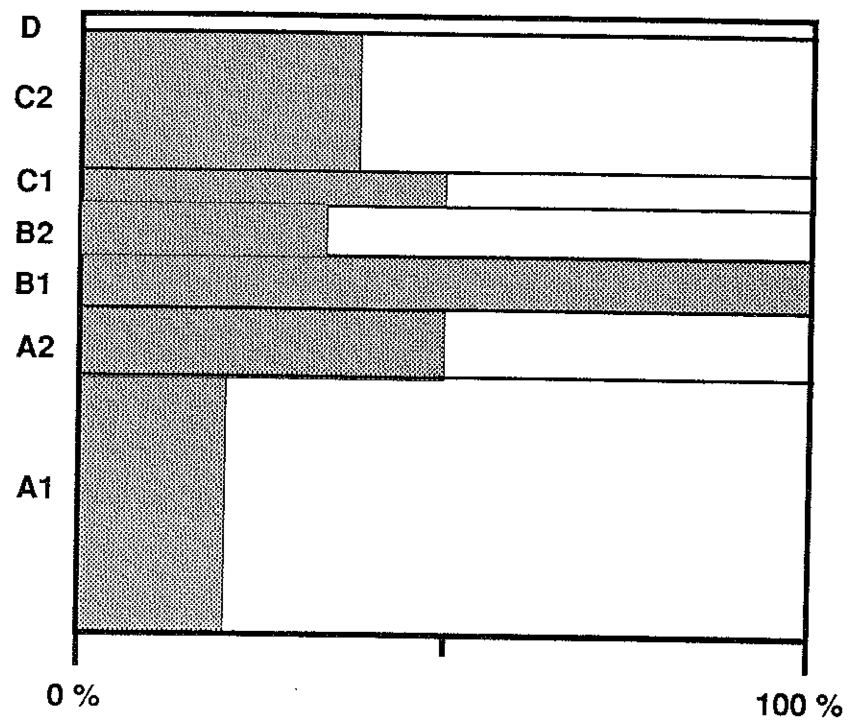
referring to: QUESTION 5.e:
Do you use purchased vector data,
without feature codes?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1m (responses to the First Questionnaire)

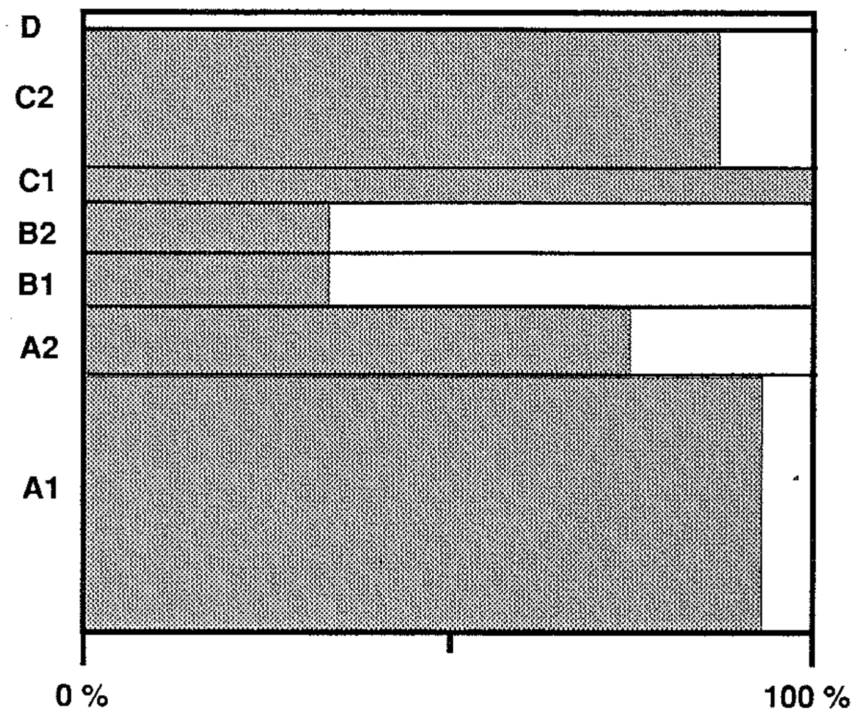
referring to: QUESTION 5.f:
Do you use purchased vector data,
with feature codes?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1n (responses to the First Questionnaire)

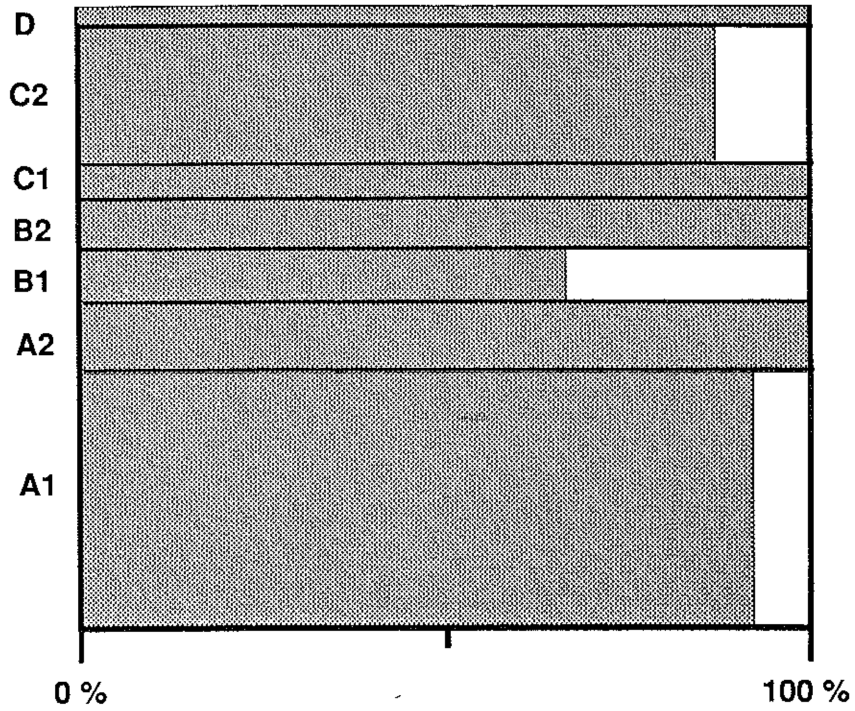
referring to: QUESTION 6:
If you use automatic digitized raster data,
do you vectorize raster data?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

FIGURE 1o (responses to the First Questionnaire)

referring to: QUESTION 7:
If you use vectorized or vector data,
do you feature code the data?



A1 National Mapping Organisations
A2 Commercial Map/Atlas Publishers
B1 Utilities National/Regional
B2 Municipalities
C1 Hardware manufacturers with associated commercial data/software bureaux
C2 Independent Commercial Data/software bureaux
D Academic/Research institutes

A4.2 THE SECOND QUESTIONNAIRE

The second questionnaire (see Appendix B for a sample) was sent only to the respondents of the first questionnaire who indicated a willingness to cooperate further, and who used automatic digitizing. This questionnaire was designed to determine how, in the production process, scanners are used by the organisations that have them.

From eight respondents the findings were that scanners are, as yet, little used in Europe for utilities, cadastral or other large scale mapping, or in situations where the scanning resulted in features later added to a graphic database (i.e. linked to the relations of a Data Base Management System). Organisations using scanners as a regular part of their production generally scanned specially prepared documents - if they did not have the production separates. The European market is dominated by the hardware of the Scitex and Sysscan companies, but although less dominant Optigraphics hardware gave a very positive reaction. Overall organisations have found that it is beneficial to use a scanner, but that problems still remain in the areas of character recognition, establishing the scanner parameters, inability to process the scanned data speedily, and poor documentation in operating manuals.

We hoped to pick up a few 'tips and tricks' from the questionnaires, but these may possibly be regarded as proprietorial and few were forthcoming. One organisation recommended transferring all images to highly reflective white opaque film before scanning, and another that sharp junctions (which cause vectorising problems) be completely removed.

Figure 2 summarises the responses to Questionnaire 2. Sometimes responses to two questions are combined - for example the questions about increased production (Appendix B, 3.2 D) and faster production (Appendix B, 3.2 F) produced exactly the same replies. In this case, this led to a suspicion that the respondents had considered the questions to be asking the same thing, and so the responses have been summarised in FIGURE 2 as 'improved productivity'.

FIGURE 2 SCANNERS IN PRODUCTION

2a THE USE OF SCANNERS IN PRODUCTION

PRODUCTION TASK/PRODUCT	% ORGANISATIONS USING SCANNER IN TASK
Production & Development	100
Development only	0
Thematic map products	87
Topographic map products	75
Atlas products	50
Cadastral map products	13
Utility map products	13
Vector data + Feature code products	87
Raster data + Feature code products	38
Vector data + No feature code products	25
Raster data + No feature code products	63
Vector data + Linked database products	38
DTM	13
Input from pencilled paper field documents	13
Input from crayoned paper field documents	25
Input from inked paper field documents	38
Input from pencilled polyester field documents	13
Input from crayoned polyester field documents	25
Input from inked polyester field documents	50
Input from pencilled paper interpretation docs.	13
Input from crayoned paper interpretation docs.	13
Input from inked paper interpretation docs.	25
Input from pencil polyester interpretation docs.	13
Input from crayoned polyester interpretation docs	13
Input from ink polyester interpretation docs.	38
Input from printed monochrome paper maps	38
Input from printed monochrome polyester maps	50
Input from printed colour paper maps	25
Input from production separates (polyester)	87
Input from production pulls (paper)	25
Input from specially prepared documents	63
Input from large scale (>1:5000) maps	38
Input from medium scale (1:5000-100000) maps	100
Input from small scale (<1:100000) maps	87

FIGURE 2 continued

2b THE BENEFITS/DISBENEFITS OF SCANNERS IN PRODUCTION

THE BENEFIT/DISBENEFIT	% ORGANISATIONS EXPERIENCINGTHIS BENEFIT/DISBENEFIT
Much financial benefit	12
Some financial benefit	63
No financial benefit	24
Much increased staff interest	38
Some increased staff interest	24
No increased staff interest	38
Much wider market	38
Some wider market	38
No wider market	24
Much improved productivity	24
Some improved productivity	63
No improved productivity	12
More scanner use now than expected	50
Same scanner use now as expected	12
Less scanner use now than expected	38
Increased scanner use will bring problems	50
Increased scanner use will bring no problems	50
Organisations have solved some scanning problems	38
Organisations have not solved any scanning problems	68
Organisations will get manufacturer to solve problems	25
Organisations will solve problems-in-house	25
Organisations will get third-party to solve problems	13

A4.3 THE THIRD QUESTIONNAIRE

The third questionnaire (also referred to as the template) consisted of seventeen sample map fragments which present particular scanning problems. Samples of these are found in Appendix C (but it should be noted that in this report only photocopied representations of the template are available). These were taken by working group members to organisations in their own countries for discussion. It was hoped that by doing this remaining problems in feature detection and vectorizing would be found.

The template examples could be differentiated from each other in terms of their SUBJECT MATERIAL, their IMAGING TECHNIQUE, and their BASE MATERIAL.

Considering the subject material the template consisted of grids, photogrammetric plots, existing topographic maps, thematic maps, utility maps, aerial photos, film separates, line symbols, area symbols, text, and hillshading.

Considering the imaging technique the template consisted of line-art, screened images, or continuous tone images; black and white, or colour images; photographic emulsion, offset print, diazo copy, photocopier tones, or pencil images; and, positive, or negative images.

Considering the base material the template consisted of stable materials such as film; unstable materials such as paper; clear, translucent, or opaque materials.

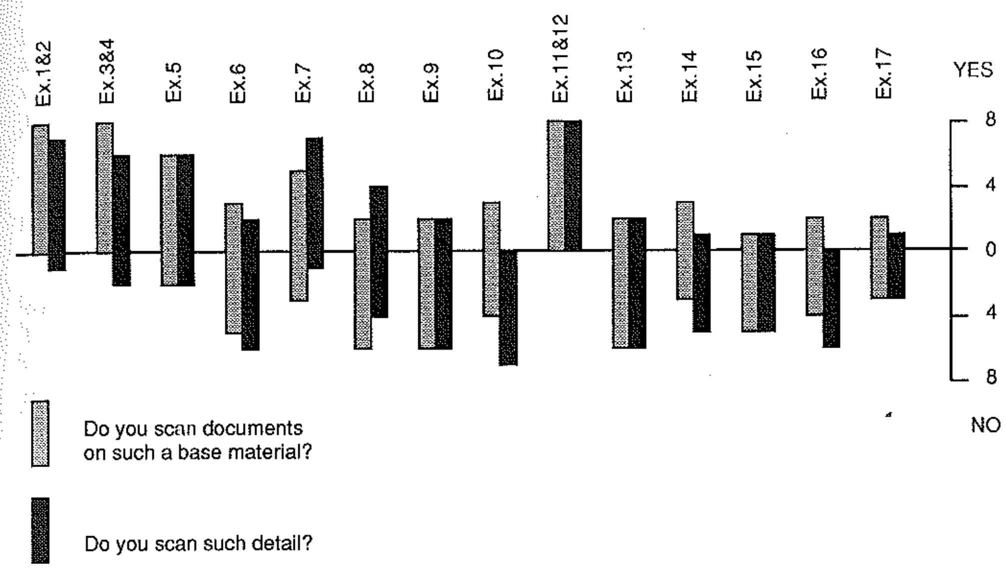
Firstly, for each sample in the template the participating organisations were asked whether:
1. they scanned documents on such a base material;
2. they scanned such detail (subject matter);
3. what problems they had with such material; and,
4. what they did with the data gathered from such material.

Secondly, for each sample in the template specific questions were also asked. For example, for a fairly complex topographic sample which had intermingled contours and rock symbolisation, the participants were asked how they extracted the contours from such a map and whether the rock symbolisation was useful in their information base. These specific questions can also be found in the Appendix C, with the relevant sample maps. National representatives have copies of all answers to the questionnaires, however a summary of these answers follows - for the reader of this report.

As stated above, associated with each example in the 'template' are four general questions. The first two had very simple answers: yes or no. These answers are summarised in the diagram (FIGURE 3) below.

FIGURE 3 SUMMARY OF ANSWERS TO QUESTIONS 1 and 2

Responses to the general questions on utilisation of the seventeen template examples



The responses to the remaining questions cannot be so crisply summarised!

There were found to be five groups of problems. These were problems associated with:

1. preparing the documents;
2. scanning the documents;
3. raster processing of the captured data;
4. vectorizing; and,
5. automatic feature coding.

These will be dealt with in turn, and problems, solutions suggested in the questionnaires, and further questions arising shall be outlined.

A4.3.1 PROBLEMS ASSOCIATED WITH DOCUMENT PREPARATION

Respondents clearly preferred clean, simple, line maps with a uniform density printed in black ink, drawn in black ink, or scribed. Pencilled originals required an uneconomic amount of post-processing. It would be useful to know whether in such cases redrafting in ink followed by scanning, or manual digitizing is invariably more efficient?

For documents needing some preparation prior to scanning, techniques involved included photographically producing colour separates; photographically copying the material onto a clear white base; enhancing lines, erasing spots and superfluous text manually; and erasing dense detail. These techniques were nevertheless found to be time consuming and there are always some documents which are just too bad to scan - no preparation can help.

A4.3.2 PROBLEMS ASSOCIATED WITH DOCUMENT SCANNING

Problems arise when scanning clear film and using reflected light - the film may not be very clear, and the black image not very black or rather 'grey', so the contrast between the two is inadequate. One solution is to place the film on white paper, but then some scanheads may 'see' a shadow because of the angular difference between the incident and reflected light. The shadow will give problems later during vectorizing. Of course this shadow does not arise if the film and paper are completely in contact, but in practice this cannot be achieved. It seems that with transmitted light the contrast problem is not so serious when the originals are 'grey'.

Coloured originals give problems for binary scanners - blue and text on some coloured backgrounds may be undetectable.

Fine detail gives problems. This detail may either be in the form of close detail (e.g. dense contours) or detailed symbols. In either case there is a tendency for the scanned pixels to be clustered.

A4.3.3 PROBLEMS ASSOCIATED WITH RASTER PROCESSING THE DATA

It is common for some raster processing to be carried out prior to vectorizing. This is mostly noise removal, but may result in fine detail, such as spot heights, rock symbols, or parts of text characters being lost.

Data may be geometrically transformed in raster form, but some systems seem to have inadequate software for this. This type of processing is time consuming - but is

frequently performed when small scanners are used which can only scan fragments of maps. The transformed fragments are then combined to form a file representing the original map. This is then vectorised.

A4.3.4 PROBLEMS ASSOCIATED WITH VECTORISATION

There is a large number of these problems. Each is briefly described below.

Fine detail is often lost.

Lines of unequal weight which intersect result in displacement of the node representing the intersection point, the problem being especially serious if the intersection is at a sharp angle. Bridging also occurs often between lines of unequal thickness.

Text is badly vectorized, but this is assumed to be because the resolution which is adequate for most map detail is not adequate for text.

Right angled corners become mishapen.

Automatic gap closing only works if the gap is smaller than any possible correct gap between individual items. Some software has problems with connecting the gaps in pecked lines because the pecks are vectorised as diagonals. A pecked line with tortuous bends can rarely be vectorised as a continuous line.

A4.3.5 PROBLEMS ASSOCIATED WITH AUTOMATIC FEATURE CODING

Text is not easily recognised on maps because of the effect of background colour and detail.

Lines differentiated by weight are not easily distinguished, but colour and linestyle is an effective way of distinguishing linear features.

Point symbols, like text, cannot easily be recognised when they are part of a map - rather than just a clean test template.

B. PHASE I OF THE STUDY ASSESSMENT OF THE POTENTIAL OF LOW COST SCANNERS

This assessment is based on a search of the commercial and scientific literature dealing with designed capabilities of low-cost scanners; attendance at exhibitions and seminars dealing with scanning and desktop publishing; testing the feature detection capabilities of three low cost scanners (Agfa, Canon, and Scan-Cad); the geometric accuracy of two of these (Scan-Cad, and the Agfa Focus Scanner) compared to established accuracies; and experiences with some vectorizing software, both low-cost and established.

As ITC has a Scan-Cad attachment in house it was able to perform extensive tests on this device. Also as the Research Assistant had professional contact with an organisation using an Agfa Focus scanner we were able to perform several tests on this device too.

Our initial work involved searching the computer graphics, especially personal computer, literature; we have continued to do this - which is necessary in a fast changing technology. During the early stages of this literature search the distinctive features of the Scan-Cad attachment of Houston Instrument (large format as its being an attachment of a popular plotter) appealed to us, so with the cooperation of its Dutch supplier we borrowed, leased, then bought the attachment. All our tests have confirmed the acceptability of this device to mapping. However during our literature search it also became clear that the Desk Top Publishing business was spawning another type of low cost scanner - a small format flat-bed scanner for document pages. The Agfa Focus scanner was one of these. This too performed well, both in terms of information retrieval and geometry - its obvious disadvantage being that one must fragment one's map for scanning. More detailed descriptions of our findings follow.

B1 LOW COST SCANNERS, A SEARCH OF THE LITERATURE

According to Haldrup [HALDRUP,1986] there are six steps in the automatic digitizing of cartographic objects. These are:

1. Preparation of the 'model' (base material, image, control, size);
2. Rasterdata capture;
3. Raster edit;
4. Vectorizing;
5. Vector edit;
6. Object Recognition.

In this section of the report we will concentrate on step 2 (capture of raster data), thus low-cost scanners will be compared on their ability to capture raster data.

Work published by Lauenstein et al., [LAUENSTEIN.1987] indicated that an assessment of scanners, as devices capturing raster data, be based on the following (with some small modifications introduced by ourselves):

General Topics

1. The name and type of the scanner.
2. The existence of an identical scanner with a different brand name.
3. Manufacturer, or a distributor.
4. Other hardware required with the scanner.
5. Software required with the scanner.
6. Cost of the scanner device.

Original Document

7. Maximum size of original document (x,y, and z).
8. Maximum size of scanned image.
9. Base material - is flexibility a requirement
   - must base material be transparent/translucent/opaque
10. Image type - line/halftone/continuous tone
   - colour/monochrome
   - offset print/diazocopy/photocopy/pencil/ink/photo/scribecoat
   - positive/negative

Scanning Operation

11. Feed - manual/automatic.
12. Support - friction wheel/vacuum/metal clamps/glass plate.
13. Light - laser/xenon/white.
14. Geometric calibration - during scanning/after scanning.
15a. Thresholding - threshold coding or greyscale coding
15b. Code setting - manual or automatic
16. Colour filter requirements.

Scanner Performance

17. Resolution (in microns and dots per inch)
18. No. of resolution steps which can be selected.
19. Scantime.
20. No. of detectable greylevels (+ No. of pseudo-greylevels, dithered patterns)
21. No. of detectable colours.
22. Accuracy.

Data Format

23. Output format
24. Pixel data - binary/greylevel/colour

Problems

25. Maintenance.
26. Mechanical wear.
27. Selftests.
28. Document positioning.
29. Runtime reliability.

Obviously the last five points (Problems) cannot be considered without practical experience with the equipment, so will only be dealt with later with respect to the one scanner with which we were able to gain experience. Also Point 10 cannot really be addressed without experience on the system. Commercial literature is often quite optimistic about the images which can be scanned; reality is different. Furthermore as we are concentrating on low-cost scanners, essentially micro-hosted, some other points cannot be considered. These are discussed prior to the tabular summary (TABLE 1), in the following paragraphs.

There are some low-cost scanners (up to Dfl 20,000) which dominate the market in that they are the scanner components of several systems. They may appear with new names. This is the case with some of the Agfa, CANON, and MICROTEK scanners, but it is not always clear in the commercial literature when this has happened. This situation is referred to in point 2 above, and we have addressed it in the table by grouping several scanners that seem to us to be almost identical.

An indication of the cost of the scanner device is mentioned in TABLE 1. This includes scanner software and all necessary cables and interfaces. This does not include software for editing or optical character recognition.

We found several types of low cost scanners. The most simple, cheap and gadget-like type of scanner is the so-called Handy scanner. Such a scanner is a small hand-held device. When dragged over the original, it can scan a strip that is only 6 or 10 cm wide. Slightly more stable is the roller-type scanner. In a roller-type scanner the original is moved across the scanhead by friction feed. This technique is very similar to that of TeleFAX machines and printers. Both Epson and ThunderScan offer accessories to convert a matrix printer into a simple scanner. As already mentioned, Houston Instrument offers an accessory that converts their belt-bed plotter into a simple but effective large-size scanner. Only for this second type of low cost scanner flexibility of the original is required. The third type of low-cost scanner we found is the flatbed scanner. In most flatbed scanners the scanhead is positioned under the original, which lies flat on a glassplate (much like in a photocopy-machine). In some flatbed scanners the scanhead is mounted on an arm that is positioned over the original. We have chosen not to consider video cameras, the fourth type of low-cost scanners, in this study. The reason for this is that other researchers involved (both in other OEEPE Working Groups and in ITC) in image processing are looking at these as scanner components, and the needs of workers in the topographic mapping area are to some extent satisfied with binary images - either scanned line maps, or scanned production separates. Binary devices are very simple, and can therefore be very cheap. It was thought worth studying them on their own. At a later date the results of the video camera study can be compared with a study of more elegant image scanners. Also at a later stage some mid-cost scanners (costing about US$50,000.00), such as the new Scitex Smart Scanner and Compugraphic's MCS 2000 may be considered.

More expensive scanners sometimes provide a choice between transmitted and reflected light. This choice, revealing itself in the possibility of using clear material (Point 9 - basematerial: clear/translucent/opaque) is not often available with low-cost scanners, and when carried out may produce 'shadows' - see FIGURE 4b where there is some evidence of this. Another aspect of the scanning operation is the feeding of the document into the scanner. For some low-cost scanners additional automatic feeders may be attached. For the more expensive scanners traditionally used in mapmaking, automatic feed has not appeared - and this is probably because captured data will be transformed

to fit the ground coordinate system after vectorizing anyway - so whether or not the original document was scanned in an orthogonal position with respect to the scanner's axes is unimportant. We therefore decided not to address this (point 11).

Having said light is always reflected in low-cost scanners, it is also always white-light that is reflected in these scanners (e.g. not laser-light) (point 13).

Considering point 14, some of the more expensive scanners permit 'on the fly' geometric transformation from the scanner coordinate system to the ground coordinate system - during scanning. Because of their origins in the Graphics Industry, low-cost scanners do not have this capability. This geometric transformation software software will always have to be produced by the mapping organisation.

When considering the scanning operation, point 16 (the use of colour filters) is also not considered in this study, because in these low-cost scanners almost all machines record only black/white. The one exception we found was the SCANMASTER (Howtek) recording over 16 million colours. The Howtek SCANMASTER uses colour filters.

TABLE 1 SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 1 - GENERAL TOPICS (POINTS 1-6)

SCANNER POINT	3(manuf)	4(hware)	5(sware)	6(guilders)
1. AGFA S200 PC	AGFA	IBM micro	MacScan	8900
... AEG Olympia Scanner	AEG OLYMPIA	IBM micro	MacScan	?
... Compugraphic MCS1000 Logoscanner	Compugraphic	IBM micro	MacScan	10900
2. AGFA MacScan Flatbed	AGFA	Macintosh	MacScan	11900
3. AGFA Focus S800GS	AGFA	Macintosh IBM micro	McView+	13000
... AGFA Focus S600GS	AGFA	Macintosh IBM micro	McView+	12000
4. AGFA Focus S800	AGFA	Macintosh IBM micro	McView+	9900
... AGFA Focus S600	AGFA	Macintosh IBM micro	McView+	10000
5. Apple Scanner	Apple	Macintosh	AppleScan	5000
6. Canon Image Scanner IX-12	Canon	IBM	GEM-Scan	5600
... Corel Sheetfed Scanner	Corel System	IBM	?	3800
... Heyden MacScan LS-300	Heyden&Son	Macintosh	MacScan	3800
... Nw.Img.Techn. MacScan	New Image Technology	Macintosh	MacScan	3200
... Pr.Gr.Syst. LS-300	Princeton Graphic Syst.	IBM	?	3100
... Canon Image Scanner IX-8	Canon	IBM	GEM-Scan	4800
7. Canon Image Scanner IX-12/F	Canon	IBM	GEM-Scan	6400
8. Contex Full Scale Scanner 3012	Computer& Busin.Mach.	IBM	CADIMAGE	25625

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 1 - GENERAL TOPICS (POINTS 1-6)

SCANNER POINT	3(manuf)	4(hware)	5(sware)	6(guilders)
9. Dest PC Scan 1000	Dest	Macintosh IBM	?	5200
10. Dest PC Scan 2000	Dest	Macintosh IBM	Publish Pac	6300
11. Epson Scanner Option Kit	Epson	IBM &Epson prtr.	EpScan	600 &printer
12. Handy Scanner HS-1000	DFI	IBM	HaloDPE	500
... AMS MicroScan	Advanced Memory Syst.	IBM	Amscan	900
... Complete Hand Scanner	Paperback Direkt	IBM	Soft Stationary	700
... Handy Scanner	Handy	IBM	?	?
... GeniScan	GameWorld	IBM	ScanEdit &Dr.Halo	600
... Logitech ScanMan	Logitech	Macintosh IBM	Paintshow+	1000 600
... Marq System-mouse	Marq Techn.	IBM &MSmouse	?	1600
13. Hewlett-Packard Scanjet	Hewlett-Packard	IBM	Scanning Gallery	3000
... Corel Flatbed Scanner	Corel System	IBM	?	?
14. Houston Instrument Scan-CAD	Houston Instrument	IBM &DMP50 pltr.	Scan-CAD	6200 &plotter
15. Howtek Scanmaster	Howtek	Macintosh IBM	Sc.Appl. ScanIt	19800
... Sharp JX-450 Color Scanner	Sharp Electronics	Macintosh	Laser-Paint	22900

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 1 - GENERAL TOPICS (POINTS 1-6)

SCANNER / POINT	3(manuf)	4(hware)	5(sware)	6(guilders)
16. Microtek MS-300(A&C)	Microtek Lab.	Macintosh IBM	VersaScan+ EyeStar+	4000
... AST TurboScan	AST Research	IBM	TurboScan	3500
... Abaton Scan 300/SF	Abaton Technology	Macintosh IBM	C Scan+ Panel Scan	3900
... Datacopy JetReader	Datacopy	Macintosh IBM	MacImage WIPS	?
17. Microtek MSF-300G	Microtek Lab.	Macintosh IBM	VersaScan+ EyeStar+	8000
... Abaton Scan 300/FB	Abaton Technology	Macintosh	C Scan	4700
... Datacopy Model 730	Datacopy	Macintosh IBM	MacImage WIPS	5000
... Datacopy Model 830	Datacopy	Macintosh	MacImage	7200
18. Microtek MSF-300Q	Microtek Lab.	Macintosh	?	6700
19. Mirror Technology VisionScan	Mirror Technology	Macintosh	VisionScan	1500
... Chinon Deskscan 2000	Firdata	IBM	?	2200
20. Siemens Scanner ST-400	Siemens	Macintosh	Siemens HYGRIS	10000
21. Siemens Scanner ST-800	Siemens	IBM	Siemens HYGRIS	?
22. ThunderScan	Thunderware	Macintosh &Apple prtr.	Thunder Scan	500 &printer

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 2 - ORIGINAL DOCUMENT (POINT 7) & SCANNING OPERATION (POINT 12)

SCANNER / POINT	7(max dim)	12(support)
1. AGFA S200 PC	A4	flatbed
... AEG Olympia Scanner	A4	flatbed
... Compugraphic MCS1000 Logoscanner	A4	flatbed
2. AGFA MacScan Flatbed	A4	flatbed
3. AGFA Focus S800GS	A4	flatbed
... AGFA Focus S600GS	A4	flatbed
4. AGFA Focus S800	A4	flatbed
... AGFA Focus S600	A4	flatbed
5. Apple Scanner	legal	flatbed
6. Canon Image Scanner IX-12	21 cm wide	roller-type
... Corel Sheetfed Scanner	21 cm wide	roller-type
... Heyden MacScan LS-300	21 cm wide	roller-type
... Nw.Img.Techn. MacScan	21 cm wide	roller-type
... Pr.Gr.Syst. LS-300	21 cm wide	roller-type
... Canon Image Scanner IX-8	21 cm wide	roller-type
7. Canon Image Scanner IX-12/F	A4	flatbed
8. Contex Full Scale Scanner 3012	A0 !	roller-type

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 2 - ORIGINAL DOCUMENT (POINT 7) & SCANNING OPERATION (POINT 12)

SCANNER	POINT 7(max dim)	12(support)
9. Dest PC Scan 1000	A4	flatbed
10. Dest PC Scan 2000	21 cm wide	roller-type
11. Epson Scanner Option Kit	15/28 x 20 cm	matrix printer
12. Handy Scanner HS-1000	6.4 cm wide	hand-held
... AMS MicroScan	?	hand-held
... Complete Hand Scanner	6.4 cm wide	hand-held
... Handy Scanner	6.4 cm wide	hand-held
... GeniScan	10.5 cm wide	hand-held
... Logitech ScanMan	10.0 cm wide	hand-held
... Marq System-mouse	?	hand-held
13. Hewlett-Packard Scanjet	legal	flatbed
... Corel Flatbed Scanner	legal	flatbed
14. Houston Instrument Scan-CAD	A1/A0 !	plotter
15. Howtek Scanmaster	A3 !	flatbed
... Sharp JX-450 Color Scanner	A3 !	flatbed

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 2 - ORIGINAL DOCUMENT (POINT 7) & SCANNING OPERATION (POINT 12)

SCANNER	POINT 7(max dim)	12(support)
16. Microtek MS-300(A&C)	21 cm wide	roller-type
... AST TurboScan	21 cm wide	roller-type
... Abaton Scan 300/SF	21 cm wide	roller-type
... Datacopy JetReader	21 cm wide	roller-type
17. Microtek MSF-300G	legal	flatbed
... Abaton Scan 300/FB	A4	flatbed
... Datacopy Model 730	A4	flatbed
... Datacopy Model 830	A4	flatbed
18. Microtek MSF-300Q	legal	flatbed
19. Mirror Technology VisionScan	A4	flatbed + overhead arm
... Chinon Deskscan 2000	A4	flatbed + overhead arm
20. Siemens Scanner ST-400	A4	flatbed
21. Siemens Scanner ST-800	A4	flatbed
22. ThunderScan	21 cm wide	matrix printer

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 3 - SCANNER PERFORMANCE

SCANNER	POINT	17(resolution microns/dpi.)	18(no. of resn.steps)
1. AGFA S200 PC		64-254 microns /400-100 dpi.	4
... AEG Olympia Scanner		85 microns /300 dpi.	1
... Compugraphic MCS1000 Logoscanner		85 microns /300 dpi.	1
2. AGFA MacScan Flatbed		63- 85 microns /406-300 dpi.	?
3. AGFA Focus S800GS		32-254 microns /800-100 dpi.	13
... AGFA Focus S600GS		42-340 microns /600- 75 dpi.	13
4. AGFA Focus S800		32-254 microns /800-100 dpi	13
... AGFA Focus S600		42-340 microns /600- 75 dpi.	13
5. Apple Scanner		85-340 microns /300- 75 dpi.	5
6. Canon Image Scanner IX-12		85-340 microns /300- 75 dpi.	4
... Corel Sheetfed Scanner		85 microns /300 dpi.	1
... Heyden MacScan LS-300		85-340 microns /300- 75 dpi.	4
... Nw.Img.Techn. MacScan		85-340 microns /300- 75 dpi.	4
... Pr.Gr.Syst. LS-300		85 microns /300 dpi.	1
... Canon Image Scanner IX-8		127 microns /200 dpi.	1
7. Canon Image Scanner IX-12/F		85-340 microns /300- 75 dpi.	4
8. Contex Full Scale Scanner 3012		85-340 microns /300- 75 dpi.	4

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 3 - SCANNER PERFORMANCE

SCANNER	POINT	17(resolution microns/dpi.)	18(no. of resn.steps)
9. Dest PC Scan 1000		85-340 microns /300- 75 dpi.	?
10. Dest PC Scan 2000		85-340 microns /300- 75 dpi.	?
11. Epson Scanner Option Kit		282-353 microns / 90- 72 dpi.	2
12. Handy Scanner HS-1000		127 microns /200 dpi.	1
... AMS MicroScan		127 microns /200 dpi.	1
... Complete Hand Scanner		127 microns /200 dpi.	1
... Handy Scanner		127 microns /200 dpi.	1
... GeniScan		127 microns /200 dpi.	1
... Logitech ScanMan		127-254 microns /200-100 dpi.	2
... Marq System-mouse		85 microns /300 dpi.	1
13. Hewlett-Packard Scanjet		85-340 microns /300- 75 dpi.	?
... Corel Flatbed Scanner		85 microns /300 dpi.	1
14. Houston Instrument Scan-CAD		127 microns /200 dpi.	1
15. Howtek Scanmaster		85-340 microns /300- 75 dpi.	4
... Sharp JX-450 Color Scanner		85-340 microns /300- 75 dpi.	5

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 3 - SCANNER PERFORMANCE

SCANNER	POINT	17(resolution microns/dpi.)	18(no. of resn.steps)
16. Microtek MS-300(A&C)		85 microns /300 dpi.	1
... AST TurboScan		85 microns /300 dpi	1
... Abaton Scan 300/SF		85 microns /300 dpi	1
... Datacopy JetReader		85 microns /300 dpi	1
17. Microtek MSF-300G		85-340 microns /300- 75 dpi.	6
... Abaton Scan 300/FB		85 microns /300 dpi	1
... Datacopy Model 730		85-340 microns /300- 75 dpi.	8
... Datacopy Model 830		85-340 microns /300- 75 dpi.	8
18. Microtek MSF-300Q		85-340 microns /300- 75 dpi.	?
19. Mirror Technology VisionScan		127-254 microns /200-100 dpi.	4
... Chinon Deskscan 2000		127 microns /200 dpi.	1
20. Siemens Scanner ST-400		64-127 microns /400-200 dpi.	3
21. Siemens Scanner ST-800		32-127 microns /800-200 dpi.	5
22. ThunderScan		127-340 microns /200- 75 dpi.	3

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 4 - SCANNER PERFORMANCE (POINT 20) & DATA FORMAT (POINT 23)

SCANNER	POINT	20(greylevels + dithered patt.)	23(output format)
1. AGFA S200 PC		2 dithered 64	PAINT,PICT,EPS
... AEG Olympia Scanner		2 dithered 64	?
... Compugraphic MCS1000 Logoscanner		2 dithered 64	?
2. AGFA MacScan Flatbed		2	PAINT,PICT,EPS
3. AGFA Focus S800GS		64	PAINT,PICT,EPS,TIFF
... AGFA Focus S600GS		64	PAINT,PICT,EPS,TIFF
4. AGFA Focus S800		2	PAINT,PICT,EPS,TIFF
... AGFA Focus S600		2	PAINT,PICT,EPS,TIFF
5. Apple Scanner		16	PAINT,PICT,TIFF
6. Canon Image Scanner IX-12		2 dithered 32	.IMG, .TIF
... Corel Sheetfed Scanner		2 dithered 32	?
... Heyden MacScan LS-300		16	PAINT,PICT,EPS,TIFF, FOTO,SuperPaint
... Nw.Img.Techn. MacScan		?	?
... Pr.Gr.Syst. LS-300		2	.PCX
... Canon Image Scanner IX-8		2	.IMG, .TIF
7. Canon Image Scanner IX-12/F		2 dithered 32	.IMG, .TIF
8. Contex Full Scale Scanner 3012		2 dithered 32	.IMG, .TIF, .RLC, .Scn, .GTX, .FAX, .PCX

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 4 - SCANNER PERFORMANCE (POINT 20) & DATA FORMAT (POINT 23)

SCANNER POINT	20(greylevels + dithered patt.)	23(output format)
9. Dest PC Scan 1000	16	PAINT,PICT,TIFF
10. Dest PC Scan 2000	256	PAINT,PICT,EPS,TIFF, FOTO, SuperPaint
11. Epson Scanner Option Kit	2	.MSP, Bload
12. Handy Scanner HS-1000	2 dithered 16	.MSP, .GEM ?? .TIF, .IMG
... AMS MicroScan	2	.MSP, .IMG, .TIF
... Complete Hand Scanner	2	?
... Handy Scanner	2	?
... GeniScan	2	?
... Logitech ScanMan	2	Windows clipboard
... Marq System-mouse	?	?
13. Hewlett-Packard Scanjet	16	.TIF
... Corel Flatbed Scanner	?	?
14. Houston Instrument Scan-CAD	2	.RAS, .CUT
15. Howtek Scanmaster	64 per colour!	.TIF, TIFF e.o.
... Sharp JX-450 Color Scanner	64 per colour!	TIFF e.o.

TABLE 1 (cont.) SUMMARY of the CAPABILITIES of LOW-COST SCANNERS

Part 4 - SCANNER PERFORMANCE (POINT 20) & DATA FORMAT (POINT 23)

SCANNER POINT	20(greylevels + dithered patt.)	23(output format)
16. Microtek MS-300(A&C)	2 dithered 64	?
... AST TurboScan	2 dithered 52	PAINT,EPS,FOTO
... Abaton Scan 300/SF	16	?
... Datacopy JetReader	2 dithered 16	.PCX, .TIF PAINT,TIFF,SuperPaint
17. Microtek MSF-300G	256 ?	PAINT,PICT,EPS,TIFF
... Abaton Scan 300/FB	16 ?	?
... Datacopy Model 730	2 dithered 16	.PCX, .TIF PAINT,PICT,EPS,RIFF,TIFF
... Datacopy Model 830	64	PAINT,PICT,EPS,RIFF,TIFF
18. Microtek MSF-300Q	64	PAINT,PICT,EPS,TIFF
19. Mirror Technology VisionScan	2	PAINT,PICT,TIFF
... Chinon Deskscan 2000	?	?
20. Siemens Scanner ST-400	64	PAINT,PICT,EPSF,TIFF
21. Siemens Scanner ST-800	64	.EPS, .TIF
22. ThunderScan	2 dithered 16	PAINT,PICT,EPSF, ThunderScan

Of course the information in TABLE 1 is not complete, the "?'s" show that. In addition to the scanners referred to in TABLE 1 some information was obtained about other low-cost scanners (Advanced Memory Systems, ALOT, CompuScan Model PCS235, Computer Aided Technology's High Speed Scanner, another small format scanner from CONTEX, Datacopy Model 840 and 840I, two EIT Personal scanners, yet another Hand Scanner, IBM scanners 3117 and 3118, Infotek's MacdigII, Intelligent Optics' IOC Reader, I-SYS Model 101, Kurzweil, Laser Connection IS-300, three more scanners from New Image Technology, OAZ Scan, Packard Bell DA 300-S Intelligent Image Scanner, Panasonic FX-RS505, Pentax SB-A4301 Image Scanner, PixiScanner, Ricoh IS-30, Saba Technologies' Page Reader, ScanPerfect, a new Siemesns scanner, Spectrafax 200 (colour), Taxan Crystal Scan, Taxan TX300, Totec Proscan TO-5050, Truvel's Truscan, Xerox), but there was insufficient information on these to add them to our list. We also obtained information on some video scanners (ORKIS CAMESCAN, PROVIZ Video digitizer, Quick Capture, PCVision Plus, EASY SCANNER, VIDEO SCANNER, Neotech Image Grabber, Scion Image Capture Board) but did not include them in the survey, as already mentioned.

The Scan-Cad attachment was chosen by us for acquisition by ITC and for our study simply because its large format (A0 or A1) was thought to be convenient for mapmakers, and also because it is driven by a Houston Instrument DMP52 plotter which had been selected as one of the peripherals to be used in the LIS/Cadastre course at ITC. Also Houston Instrument, Belgium and its Dutch distributor Geveke Electronics was interested in our study and loaned us the scanner for some months (we have since bought it). Its more detailed characteristics will be discussed later.

B1.1 RASTER FORMATS

Point 23 of TABLE 1 dealt with output formats. This is an important item when considering devices which may be added to existing computer assisted map production systems. In the literature search for low-cost scanners we came across a very large number of different file-formats. Almost every scanner has its own proprietary format to store its output of raster files. Fortunately some formats emerge as generally accepted standards. For simple binary files (only black&white) the most popular raster formats in the IBM microcomputer environment are .DHP (Dr.Halo Paint), .IMG (GEM Paint), and .MSP (MicroSoft Paint under MSWindows); in the Apple Macintosh environment PAINT and PICT are the most popular. For raster files that include colour and grey-scale information the most popular raster formats seems to be .PCX (PC Paint) and .TIF in IBM micros, and PICT II, PostScript, EPSF and TIFF in Apple Macintosh. Some information on these formats follows.

BLOAD is the old IBM BASIC graphic format.

.CUT is a device independent raster file format. It can be loaded into the MS-Windows Clipboard. It can be imported to Hi-Scan and other raster editing programs, see [HOUSTON INSTRUMENT,1987b] for further information.

.DHP is the Dr.Halo Paint file format.

EPSF (Encapsulated PostScript Format) can handle colour. This file format consists of two parts:

1. The first part is a description of the image in PostScript format. It can handle raster files as well as vector files. As a result of this the image can be printed on a PostScript-printer.
2. The second part is a description of the bitmapped image for display on the monitor. In Macintosh this second part is in PICT or QuickDraw format; in IBM this second part is in .TIF format. (Some applications omit this second part.) EPSF files from a scanned image are up to four times as large as Packbits compressed TIFF.

Grey can be stored and output, but an image, once saved in EPSF format, can no longer readily be manipulated at the dot or sub- dot level. Although PostScript recognises and can pass on grey values, it does not do so with a cellular method - the smallest grey-holding entity known to PostScript is the Path [MACUSER,p.82,1989].

.FAX is the raster input file for the program GammaFAX (from Gamma Technology).

FOTO file format is an older file-format for PageMaker 1.2; it is a raster format with a resolution of 300 dpi, and it can store greyscale information as dithered patterns. It was the earliest "greyscale" format on the Macintosh, but can only be recognized by PageMaker 1.2

GEM Paint file is the same as the .IMG file

Grey PostScript (on Macintosh) is the "PostScript language amended to incorporate greyscale information as well as all other routines" [DTP,1988] and is said to produce very big files.

.IMG is the same as GEM-Paint, and GEM-Scan. It is the native bit-mapped format of GEM (Graphics Environment Manager), and it is used for Ventura Publisher.

.IMG from Scan-CAD seems to be different from other .IMG files. Scan-CAD can convert its .RAS file to a file with extension .IMG The datablock of this ScanCAD .IMG file consists of a compressed description of the entire image; for further details see [HOUSTON INSTRUMENT, 1987b].

Laserbits (on Macintosh)is a 300 dpi. sub-level in SuperPaint raster format.

Laser Prep PostScript (on Macintosh) is a PostScript page description (raster and vector) that is sent ('downloaded') to the Laserprinter. "This set of PostScript procedures is developed by Apple to bridge the gap between QuickDraw, used in Macintoshes to display images on the screen, and PostScript, the language of the LaserWriter" [MACUSER, p.178, 1988b] Laser Prep PostScript can be converted to Illustrator PostScript by the program Xris-Xros.

.MSP or MicroSoftPaint (under MS Windows) can be used in Aldus Pagemaker 3.0 PC

Pagemaker 1.2 format is the same as the FOTO format.

PAINT (or PNTG) is the proprietary format of MacPaint by Apple. The Paint file format for a bitmapped image is 576 by 720 pixels. Only black and white is recorded, and no greylevels. Resolution is 72 dpi. MacPaint files can be imported by almost all graphic software on Macintosh and also by Ventura and PC PageMaker 3.0 on IBM

.PCL is the raster format of the Hewlett-Packard LaserJet. Using a LaserJet printer driver all graphics on a page can be stored on disk in this .PCL format. Resolution is 300 dpi, but text cannot be handled.

.PCX is the PC-Paintbrush format from Z-Soft. ".PCX remains the dominant bit-mapped format for paint-generated color and black-and-white bit maps on the IBM. Files in .PCX format can include color and gray-scale information and can be very large." [MACUSER, p.172, 1988b]

.PIC is the format of "grabbed images" on IBM and has no resemblances to PICT!

PICT was developed by Apple. PICT can describe large images, it can handle raster information (descriptions of bit-mapped images) as well as vector information (description of objects). PICT is the most stable and widely used format for vector files on the Macintosh. However PICT is not able to handle gray-scale images or colour. The maximum resolution for the raster information is 72 dpi. PICT files that are larger than 64K can cause problems in some applications. Some software like Cricket Draw uses extended "dialects" of PICT that can not be used in other applications.

PICT II is an enhanced version of PICT. Enhanced PICT can handle colour and it supports Color QuickDraw.

PNTG See PAINT

PostScript (or PS) is an ASCII based page-description language, developed by Adobe Systems. Linotype and Apple readily accepted PostScript as a standard, and now IBM, Hewlett-Packard and others support PostScript too. PostScript is a page-description language with over 250 commands for manipulating text and graphics. PostScript always prints out the highest resolution that is available on a printer, etc. A PostScript interpreter is required to display or print the image. It converts the PS page description to a bitmap that can be displayed or printed with the raster resolution of the chosen output device. Every output-device has its own interpreter. Such an interpreter is called a RIP, Raster Image Processor. See also EPSF, Grey PostScript, Illustrator-PostScript, Laser Prep PostScript

QuickDraw is "The language used by the Macintosh to define the appearance of almost everything on the screen. It is also used to drive dot-matrix printers, such as the ImageWriter, and non- PostScript laser printers." [MACUSER, p.81, 1988a]

.RAS is a simple bitmap format, and is the output of Houston Instrument's ScanCAD. [HOUSTON INSTRUMENT,1987b]

RIFF or Raster Image File Format is the proprietary format of Letraset's Image Studio. "RIFF is the only other format apart from TIFF which will store greys at the pixel level, and appears to be optimised for imagesetter performance - centainly it appears sturdy and relatively economical alongside the somewhat chaotic state of TIFF" [MACUSER, p.82, 1989]

.RLC is the raster input format for the program CAD-Overlay.

SuperPaint (on Macintosh) is SuperPaint's proprietary format for combined raster and vector files. In the raster-editing mode two resolutions can be used: 72 dpi, as in ordinary PAINT format; or, 300 dpi, the so called LaserBits

ThunderScan files are the proprietary format of ThunderScan.

TIFF (on Macintosh) or .TIF (on IBM micro) is the Tagged Image File Format. TIFF can handle black & white, grey, colour, one-bit and multibit images and several compression schemes. It is a flexible, extensive format but it has many dialects because of different implementations on different scanners. "There are more than 17 varieties - including so-called bi-level TIFF that will not store greys - and it is still evolving" [MACUSER, p.82, 1989]. Most TIFF formats on the Macintosh are different from the .TIF files common on IBM. Three different kinds of TIFF format are: 1. CCITT Compressed TIFF. TIFF CCITT can compress a file by a factor of four but this takes some time. It is recommended for archiving. It cannot be used for scans with 6-bit grey scales. 2. Packbits which is compressed TIFF. TIFF Packbits can compress a file by half at a high speed. (default in McView) 3. Uncompressed TIFF, produces large files. "This file type was developed by Adobe, Aldus Corporation and Microsoft to handle greyscale images, usually from scanned graphics" [MACUSER, p.81, 1988a]. "TIFF is currently the most popular format for storing cellular greyscale values" [MACUSER, p.82, 1989]. Detailed information is in [MEADOW,OFFNER,et al., 1988].

B1.2 RASTER TO RASTER REFORMATTING

If the raster format of your scanner does not match the raster formats that can be recognized by your raster-editing program you will have to convert your file from one raster format to another. Our findings with regard to raster-raster conversion are summarised in this section.

AnyGraph (on IBM) from Compatible Systems is made up of two programs. The first program, AnyPC, captures output sent to an IBM Graphics Printer and stores that on disk (max. 80 dpi.). The second part, the AnyGraph translator can convert to PAINT format.

The Curator (on Macintosh) from Solutions can convert raster files in PAINT, PICT, .EPS, EPSF, PostScript, .TIF, TIFF, and Glue (Solutions' proprietary format) to raster files PAINT, PICT, .EPS, EPSF, PostScript, TIFF, and Glue format.

CUTTOCLP.EXE (on IBM), a utility that comes with HiScan, can convert raster files in .CUT format to raster files in .CLP format.

CUTTOIMG.EXE (on IBM), a utility that comes with HiScan, can convert raster files in .CUT to raster files in .IMG format.

CUTTOTIF.EXE (on IBM), a Windows-utility that comes with HiScan, can convert raster files in .CUT to raster files in .TIF format.

The Graphics Link (on IBM) from PC Quik-Art can convert raster files in PAINT, black&white .TIF, .PCX, .MSP, .IMG, Halo .CUT and other simple formats to raster files with these same set of raster formats.

Hijaak (on IBM) from InSet Systems can convert raster files to and from a number of formats: PAINT; black&white .TIF; .PCX; .MSP; .IMG; Halo .CUT and other formats such as Amiga ILBM; CompuServe .GIF; Hewlett-Packard .PCL; Inset .PIX; and Lotus .PIC (input only); .EPS (output only).

ImageStudio (on Macintosh) from Letraset is a raster-editing program that can import raster files in PAINT, many types of TIFF, FOTO, and ThunderScan format and can save them as raster files in PAINT, EPSF, one type of TIFF format, and RIFF format.

McView Plus (on Macintosh) from Agfa can convert raster files in PAINT, PICT, and TIFF format to raster files in PAINT, PICT, TIFF, TIFF Non-compressed, and EPSF format.

Reflection (on IBM) from IMSI is identical to Hijaak.

Scan-Cad (on IBM) from Houston Instruments can convert a raster file in .RAS format to a raster file in .CUT format.

.tif to tiff (on IBM) is a utility that comes with DeskPaint from Zedcor.

B1.3 VECTOR FORMATS

Several vector formats are now well established within PC graphics. Included are .DXF (the AutoCad interchange format), .MET (metafiles under MSWindows), PICT (the MacDRAW interchange format), PostScript, and EPSF (Encapsulated PostScript).

.DRW is the internal file format of Micrografx's program Designer [BYTE, pp 181-185, 1988].

DRWG "MacDraft file format, based on drawn objects; the basic principle of object oriented graphics. Although not many other applications will open these files, MacDraft also has the option of saving files in the PICT format" [MACUSER, p.81, 1988a].

.DXF, or Drawing Interchange Format (on IBM), is from AutoCAD .DXF is the industry standard interchange vector format. DXF is an ASCII format file.

EPSF was mentioned before in the section on raster formats. It can handle vector files as well as raster files.

.GEM is the GEM line art format or vector file format from GEM. It is the object oriented standard for the GEM environment used in GEM Artline & GEM Desktop Publisher.

IGES is the Initial Graphics Exchange Standard. This vector format is used on mainframes.

Illustrator-PostScript (on Macintosh) is from Adobe. Adobe is the author of full length PostScript and nontheless they also made a condensed dialect of PostScript for Illustrator.

Lotus .PIC is a vector format for business graphics from Lotus 1-2-3. It is also used by In-a-Vision, WindowsDraw, and Designer.

.MET is a Metafile, an object-oriented file format in Windows, and is also in use for plotters.

.PIC see Lotus .PIC

PICT, mentioned before in the section on raster formats, can handle vector files as well as raster files.

.PLT is the vector file format for AutoCAD plotfiles.

PostScript, mentioned before in the section on raster formats, can handle vector files as well as raster files.

QuickDraw, mentioned before in the section on raster formats, can handle vector files as well as raster files.

SuperPaint, mentioned before in the section on raster formats, can handle vector files as well as raster files.

B1.4 RASTER TO VECTOR CONVERSION

At ITC we are fortunate to have SysScan's vectorizing software. But this is not low-cost. There is some vectorizing software available for microcomputers and this is discussed in this section.

PC-based vectorisation is possible in three ways:

1. Trace a raster file manually on screen and store the result as a vector file - using tracing programs;
2. Trace a raster file semi-automatically on screen and store the result as a vector file - using interactive autotracing programs; or,
3. Apply fully automatic vectorisation - using vectorising programs.

Tracing programs offer a range of tools to trace by hand the content of a raster image. The output file is a vector file. Five examples of these programs are described below.

1. CAD-Overlay (on IBM), by Itech Inc. brings a raster image directly into an AutoCAD drawing. The input raster data format is .RLC and the output is .DXF. The result is a "hybrid drawing".
2. Designer (on IBM AT), by Micrografx, is compatible with Windows 2.0. The input raster data format is Windows Clipboard and the output is .DRW (proprietary format), .PIC, Windows metafiles .MET and (by means of a utility) .DXF
3. Freehand (on Macintosh), by Aldus, takes the input raster format as PAINT, or PICT and the output is Freehand, Illustrator PostScript, or EPSF.
4. HiScan (on IBM), by Houston Instruments. The input raster format is .CUT and the output vector formats .TRC or .MET
5. Illustrator (on IBM and Macintosh), by Adobe. The input raster format is PAINT, or PICT and the output vector format is Illustrator-PostScript, EPSF, or .EPS

Interactive autotracing programs offer a tool that traces automatically the outline of any area of black pixels in a raster file. The output file contains vector data. Three such programs are known:
1. Freehand 2.0 (on Macintosh), by Aldus
2. Illustrator88 (on Macintosh), by Adobe
3. Digital Darkroom (on Macintosh), by Silicon Beach Sofware

Vectorising programs attempt to replace all linear features represented in the raster data set by a coordinate string representing the centre line of the feature. Seven of these are described below.

1. CAD/Camera (on IBM), from Autodesk (the creator of AutoCAD). The input raster file is in .IMG format and the output vector file is in .DXF format.
2. CAD-mate (on IBM), from Microtek. The input raster file is in .IMG format and the output vector format is in .DXF format.
3. CadScan (on IBM), part of SICAD-HYGRIS from Siemens. HYGRIS ("Hybridisches Graphisches Informationssystem") offers algorithms for thinning of linear features in the raster file, for raster-vector conversion, recognition of geometric entities and for output of vector files in HPGL, IGES, SIF, .DXF and ASCII format.
4. Digital Darkroom (on Macintosh), from Silicon Beach Software
5. McView Plus (on Macintosh), from Agfa. The input raster file is in PAINT, PICT, or TIFF format. The output vector file can be in PICT, Clipboard, or Illustrator PostScript format.
6. Scan-Pro (on IBM), from American Business Computers can convert raster files in .TIF format to vector files in .DXF, IGES and PD1 format.
7. Visus Vector Image Processor (on IBM), from Visual Understanding Systems, Inc. can convert from raster files in .RAS format to vector files in PostScript, .DXF, new .DXF (polyline), DG-Bin, Gerber, Houston DM/PL, HP/GL and IGES format.

B1.5 VECTOR TO VECTOR REFORMATTING

If the vector format that was produced by your vectorising procedures does not match the vector formats that can be handled by your vector-editing program you will have to convert your file from one vector format to another. Our findings with regard to vector-vector conversion are summarised in this section.

CadMover (on Macintosh) from Kandu Software can convert vector files in IGES, .DXF, MacDraw, MiniCAD 2D & 3D, SpaceEdit, and Snap format to vector files in IGES, .DXF, MacDraw, MiniCAD, PICT, and MSC/pal format.

DrawOver (on Macintosh), a utility that comes with Adobe Illustrator88, can convert vector files in PICT to vector "artwork" files in Illustrator-PostScript format.

HALODXF.EXE (on IBM), a utility that comes with HiScan, can convert vector files in .MET format to vector files in .DXF format.

Designer (on IBM AT) comes with a utility that converts vector files in .PIC format to vector files in .DXF format.

Xris-Xros (on Macintosh) from Taylored Graphics can convert vector files in LaserPrep PostScript (intercepted and stored on disk) to vector files in Illustrator-PostScript format.

B2 TESTING LOW COST SCANNERS

As mapmakers we are concerned with two aspects of scanning. These are:

1. the completeness of the data capture;
and,
2. the geometric accuracy of the data capture.

During the study we examined both these aspects of low-cost scanners, and our findings are described in B2.1 and B2.2

B2.1 COMPLETENESS OF THE DATA CAPTURE

In this investigation three low-cost scanners were examined. These were:

1. AGFA FOCUS S800GS
2. CANON IMAGE SCANNER IX 12/F
3. HOUSTON INSTRUMENT SCAN-CAD

With all these scanners we scanned template examples (see Appendix C). In FIGURE 4 the results of scanning template example 5 are shown. These figures are hardcopy representations of the raster files derived from the scanning. There is NO attempt to make a geometric correction. These figures must ONLY be used as a guide of low-cost scanners capabilities to pick up map details. (There is a considerable affine distortion in the plots made on the Epson printer/plotter (4c and 5c) - resulting simply from the way that device was set-up. The affine distortion is nothing to do with the Scan-Cad scanner's characteristics. The finest linework (about 0.15mm thick) on the original map (that depicting land parcel boundaries on a 1:500 scale topographic map) is of poor quality. It has not been perfectly detected by any of the scanners - although the AGFA scan is almost complete. However the next thickest line (about 0.3mm) has been detected in all cases. This indicates a need for Haldrup's 'model' preparation [HALDRUP, 1986], should low cost scanners be used on maps with the finest linework.

In FIGURE 5 the results of scanning template example 11 (see Appendix C) are shown. This time again only one of the scanners (AGFA) had little problem with the finest lines. The AGFA scanning was done at 63 microns resolution - one of the highest resolutions available on low-cost scanners. However the CANON scanner was only operating at 127 microns (0.127mm) and that is also the resolution of the SCAN-CAD scanner. In this case the thinnest original linewidth was about 0.1mm. It is one of the 'rules of thumb' of scan digitizing that the minimum line width to be scanned should be at least twice the scanner resolution, and if all detail is to be detected, this seems born out by these tests. Many of the low-cost scanners do operate at 127 microns (or 200 dpi) because this is what is required by teleFAX.

FIGURE 4a TEMPLATE EXAMPLE 5 SCANNED WITH AN AGFA SCANNER

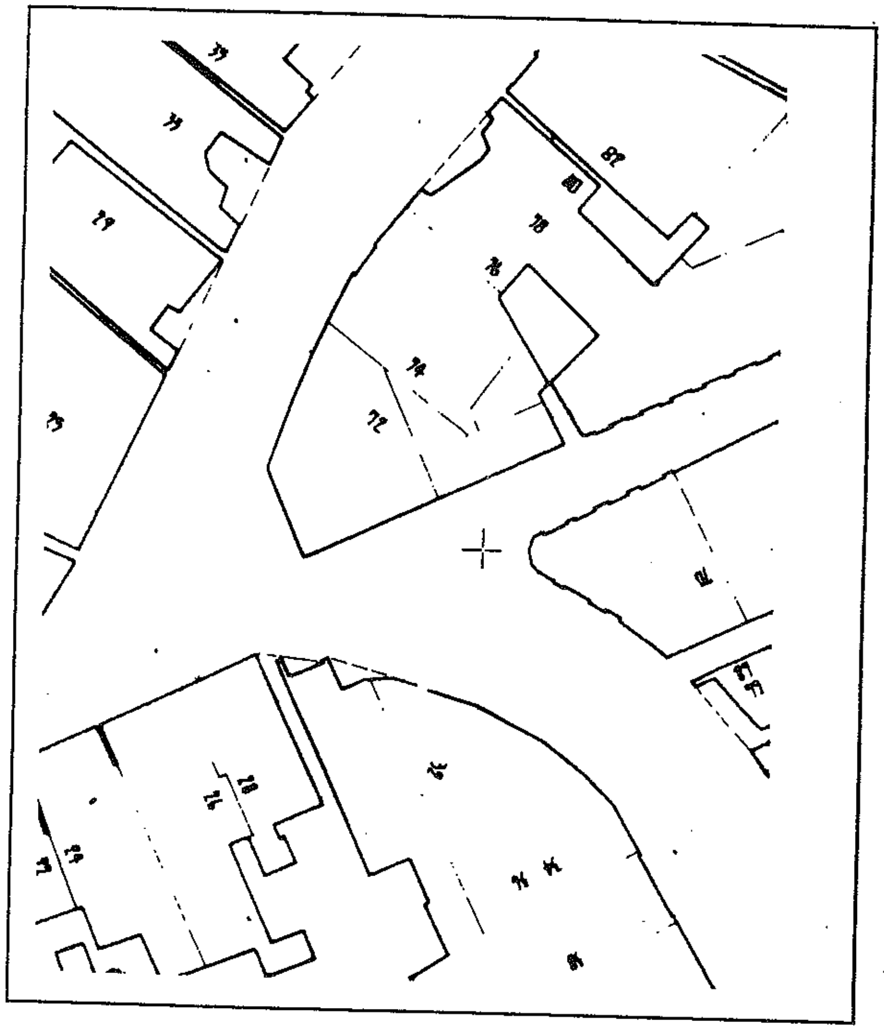


FIGURE 4b TEMPLATE EXAMPLE 5 SCANNED WITH A CANON SCANNER

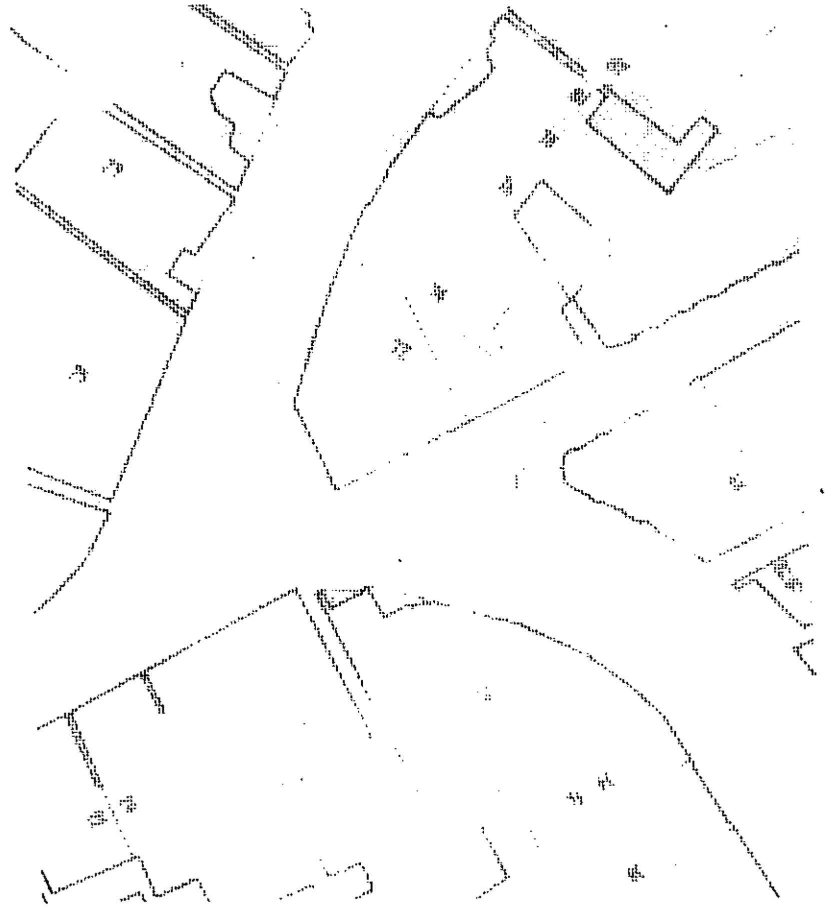


FIGURE 4c TEMPLATE EXAMPLE 5 SCANNED WITH A SCAN-CAD SCANNER

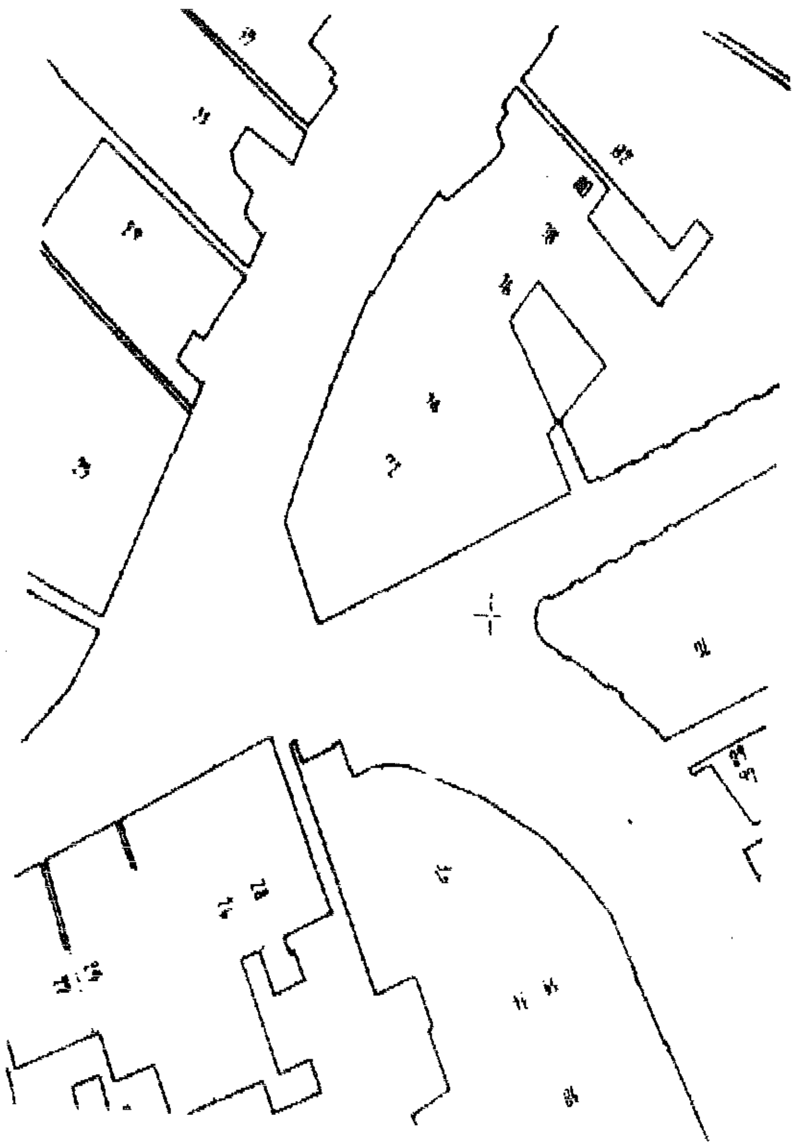


FIGURE 5a TEMPLATE EXAMPLE 11 SCANNED WITH AN AGFA SCANNER

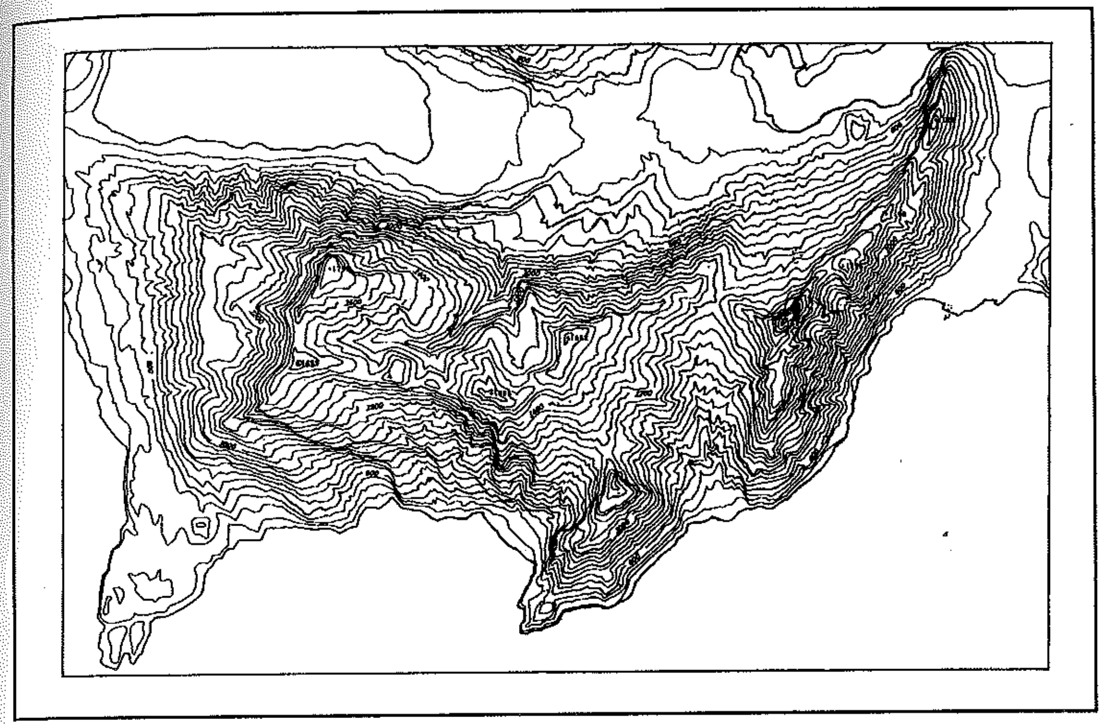


FIGURE 5b TEMPLATE EXAMPLE 11 SCANNED WITH A CANON SCANNER

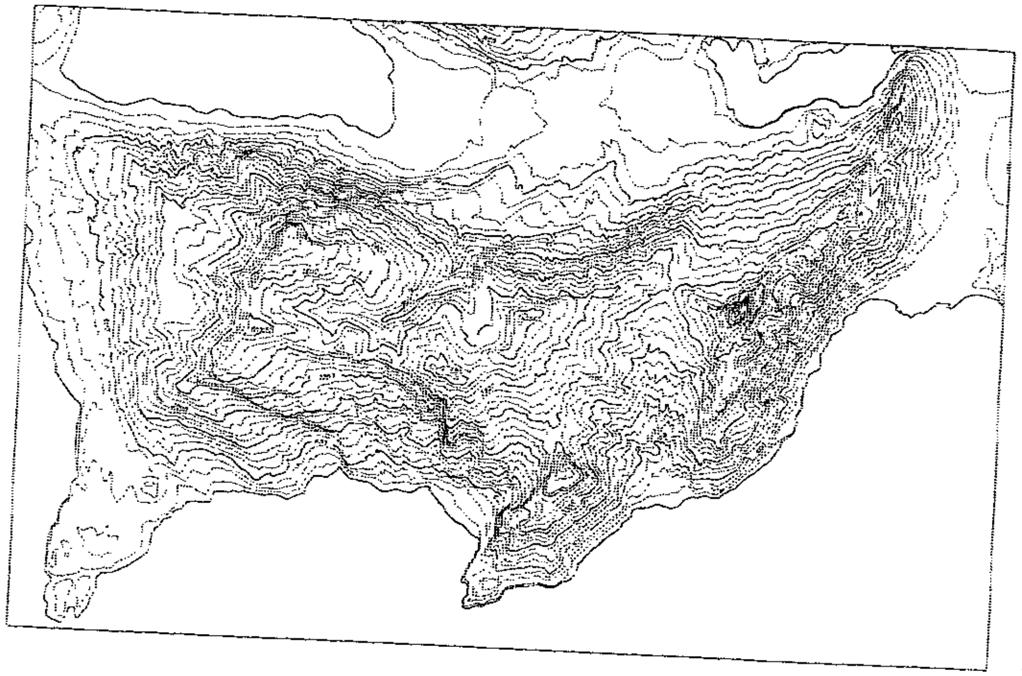
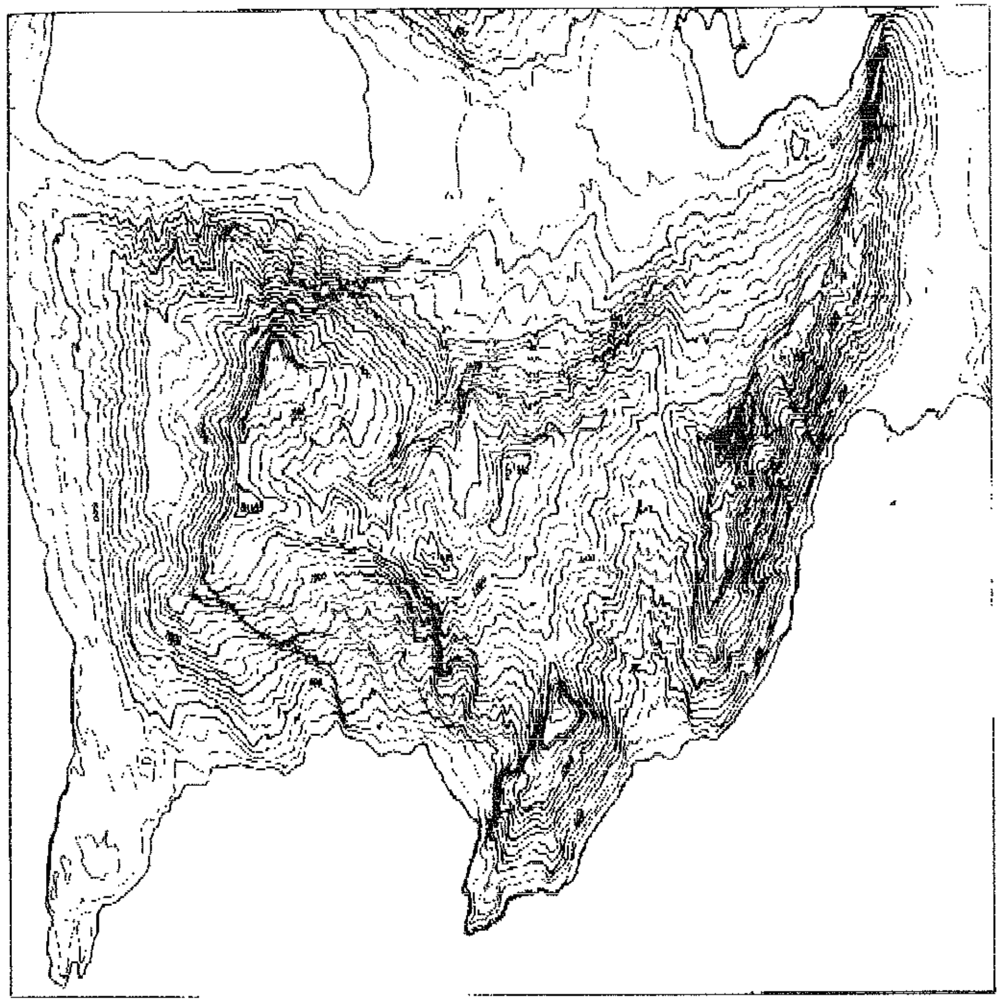


FIGURE 5c TEMPLATE EXAMPLE 11 SCANNED WITH A SCAN-CAD SCANNER



B2.2 GEOMETRIC PRECISION OF THE DATA CAPTURE

For this examination only two of the low-cost scanners were used. These were the SCAN-CAD of Houston-Instrument and the AGFA Focus S800GS. These are described in the sections 2.2.1 and 2.2.4 whereas the results of the precision tests are in sections 2.2.2, 2.2.3, and 2.2.5.

B2.2.1 HOUSTON INSTRUMENT SCAN CAD ATTACHMENT

The scanner used at ITC is the SCAN-CAD attachment of Houston Instrument. It can be attached to a plotter in the DMP 50 series of plotters. The plotter drive mechanism is used to drive the scanhead which occupies the holder usually holding a pen (see FIGURE 6). The plotter itself can draft plots up to A0 size; has a drafting resolution of 0.025mm; a stated drafting precision (repeatability or standard deviation) of 0.050mm; and a stated maximum drafting error (referred to as 'accuracy' in the operation manual) of 0.500mm. The maximum drafting error is rather high, and we have assumed that this results from the 'whiplash' of a fibre-tip pen when the plotter is operating in drafting mode. Normally one would expect the maximum error to be about 3 x standard deviation - or 0.150mm in this case. Thus on top of the constraints imposed on the SCAN-CAD accessory as a component in a scanning system, the constraints of the plotter itself, its drafting precision (0.050mm), and its resolution (0.025mm) must all be considered.

With regard to the SCAN-CAD accessory its design constraints are its resolution of 0.127mm, and a stated line detectability of 2 line pairs per milimeter of 0.178mm line width (i.e. regarding line detectability a gap of 0.644mm must exist between lines if they are to be detected as separate items). Other 'undesigned' constraints exist. These are discussed.

The scanner can detect 16 greylevels, but uses these for thresholding and it creates a binary file. A preview test allows the user to select which of the greylevels should be the threshold value. This preview test seems effective. Any features left undetected are very fine lines (see section 2.1).

FIGURE 6a THE SCAN-CAD ATTACHMENT AND ITS BELT-BED PLOTTER

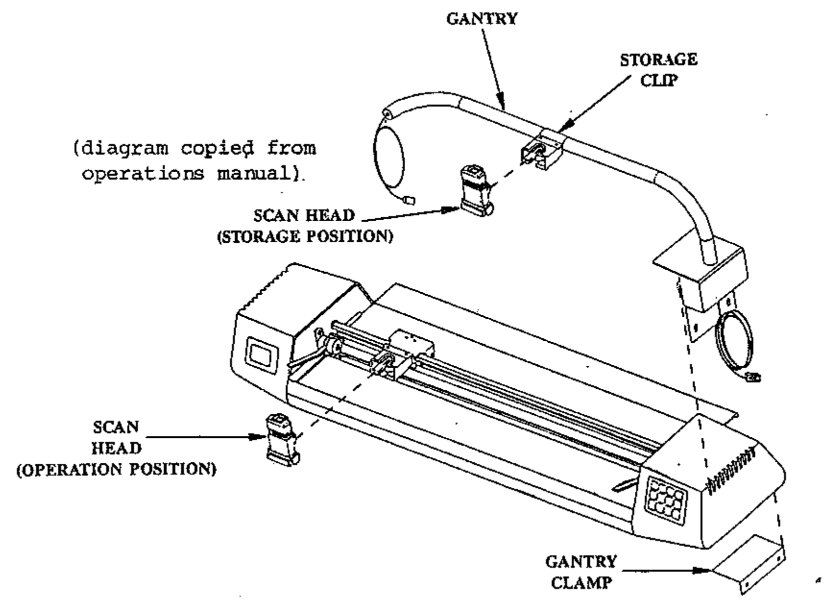


FIGURE 6b THE SCAN-CAD ATTACHMENT AT ITC - IN ACTION.
Note the modification of the roller.

During scanning the document is moved under the scanner in the x-direction, and as this happens a 120 pixel wide strip on the document is scanned (i.e. 15.24mm). As the paper moves in the x-direction it pauses at regular intervals to permit the transmission of captured data to the datafile on disk. These regular pauses are part of the system, but, as all pixel locations are represented by a sequence number which appears related to the strip number, irregular pauses caused by friction severely effect the location information associated with a pixel. Problems were experienced with irregular pauses.

At the end of scanning a strip in the x-direction the document is moved back to its start position and the scanhead moves 120 pixels (15.24mm) in the y-direction. From this position the scanning process restarts, for the next strip. It was found to take about half an hour to scan a full sheet. The resulting datafile was transferred via a networking system between IBM micros and the central VAX of ITC to ITC's SysScan system. There the data was vectorized, and underwent further examination.

Our Houston Instrument plotter at ITC is only for A1 size sheets (an A0 size is available). To use the friction feed (pinch rollers) of the plotter, the scanned map must be 59.5cm wide, so narrower documents must be mounted onto a base - thus increasing the document thickness and the chance of friction during scanning. Also associated with this is the actual means of attaching the samples to the paper base - we used masking tape and its contraction during the test caused slight buckling of one of the test sheets, this problem was revealed when we came to examine the discrepancy plots (see FIGURE 9 where the 'masking tape effect' is obvious). One solution is to modify the plotter so that the friction wheels can be repositioned over narrower documents, but we were not in a position to so modify the plotter. Another solution is to copy the document to be scanned, by photographic means, onto a film base of the required dimensions - thereby also avoiding the problem of having a rather thick source document.

The advocacy of a film document of a specified dimension as the source document does introduce other problems. The first is that the bed of the plotter is black so that there is no contrast between the features on the source document and the scanned base, but this can be overcome with a thin strip of white paint just below the scanhead's scanning location. The second is that, in order to get as crisp an image as possible, the document should be 'emulsion up'. The emulsion side of film is relatively soft and any image can be quite easily scratched. We found that friction between the document and the scanhead corrupted the image, although it was not enough friction to cause a scanning pause. Obviously, in the event that cartographic production separates are to be scanned, it cannot be permitted that the emulsion surface be destroyed in the process. An examination of the scanning process revealed that a small flattening roller which was rather smooth was in constant contact with the document, its purpose being to keep the document under the scanhead smooth, but also the plastic sides of unit holding the linear array of photosensitive cells was also in contact with the document. There seemed to be no need for this second contact, and we also thought it was this contact causing the damage to the film. The scanner was therefore modified by making the flattening roller slightly larger in circumference by adding two rings to it (see FIGURE 6b). This lifted the plastic sides of the photo-cell unit off the paper, did not effect the detection capabilities of the scanner, and prevented damage to the emulsion. This modification did not appear to introduce any extra irregular pauses.

At ITC we guessed that the designers of the SCAN-CAD attachment had had a lot of trouble with the cabling. The cable to the scanhead had to be self supporting, or its

weight would cause the scanhead to rotate in its holder, with resulting locational irregularities. This was solved by the manufacturer by having a rather rigid cable, which was brought to the location of the scanhead by a support arm. Unfortunately the support arm and the feed cable sometimes got tangled up thereby causing an irregular pause in the scanning process. It was almost impossible to stop this from happening. Eventually we discarded the support arm, and suspended the feed cable from a point in the ceiling above the scanner. This almost completely controlled irregular pausing, but the ideal solution still has to be found, especially if unattended scanning is desired.

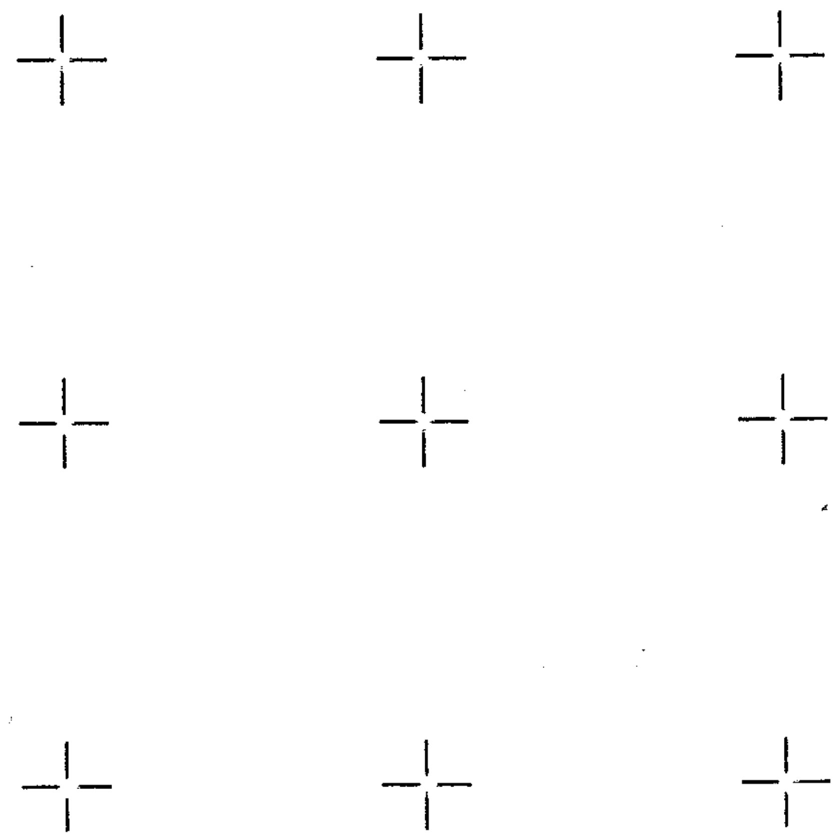
B2.2.2 GEOMETRIC TEST OF THE SCAN-CAD ATTACHMENT WITH A TEST GRID

To test the Scan-Cad attachment using a test grid the following procedure was adopted:

1. A 5cm grid of open centred crosses was plotted onto a stable base film using the photohead of ITC's Kongsberg 1612 flatbed plotter (see FIGURE 6 for a sample of part of this plot, and also 4b showing its operational use).
2. This testplot was then scanned with Scan-Cad.
3. The scanned data was transferred, as a .RAS file, to ITC's SysScan system where the cross-arms were vectorised.
4. A centrepoint for each of the cross arms was then mathematically interpolated.
5. Certain crosses were chosen as Control Points and the set of cross centre points was then transformed using these Control Points and their 'true' values as established in the original Kongsberg plotfile.
6. The transformed cross centre points were then compared with the original Kongsberg plotfile values to get discrepancies between the positions of scanned crosses and their true position.
7. These discrepancies were used as input to produce a plot showing discrepancies across the scanned map sheet, and to determine Root Mean Square Error (rmse).

(A listing of he program used in this processing may be obtained from Ms. Jane Drummond, ITC, PO Box 6, 7500AA ENSCHEDE, The Netherlands.)

FIGURE 7 A PART OF THE TEST DOCUMENT



This process was repeated for one scanning with an increasing number of control points up to nine, and then on a second scanning with nine Control Points (8-13). FIGURES 6-11 show the discrepancy plots for these scannings and data processing operations.

It should be noted that a "patchwise" adjustment was used for the 9-control point tests - that is the map was divided into four quarters, and each quarter adjusted with four control points located near each corner.

TABLE 2 shows the average discrepancy and maximum discrepancy values for the seven examined situations. It can be seen that Scan-Cad with 9 control points can produce an rmse of about 0.16 - 0.21mm and a maximum discrepancy of about 0.37 - 0.56mm. We must consider whether or not the accuracy standards achieved by the Scan-Cad are adequate for mapmakers.

TABLE 2 ACCURACY EXAMINATION OF TWO SCANNING OPERATIONS USING THE SCAN-CAD ATTACHMENT

test		RMSE (without edge crosses)	MAXIMUM ERROR (without edge crosses
1	(3 control points)	1.6mm	4.4mm
2	(3 control points)	0.7mm	2.5mm
3	(4 control points)	0.3mm	0.9mm
4	(5 control points)	0.4mm	1.0mm
5i	(9 c.p's 1 patch)	0.18mm	0.65mm
5ii	(9 c.p's 4 patches)	0.16mm	0.37mm
6	(9 c.p's 4 patches)	0.21mm	0.56mm

In [THOMPSON, p.55, 1984] the accuracies of 15 cartographic digitizing procedures were compared. The procedures included manual line following digitizing, automatic line following digitizing, and automatic scan digitizing. 'Standard errors' ranged from 0.08mm to 0.13mm, and 'Maximum Vector Errors' from 0.14mm to 0.35mm. It can be seen that the standards achieved are better than those achieved using the Scan-Cad attachment.

FIGURE 8 DISCREPANCIES WITH 3 CONTROL POINTS (CIRCLED) - RMSE=1.6mm

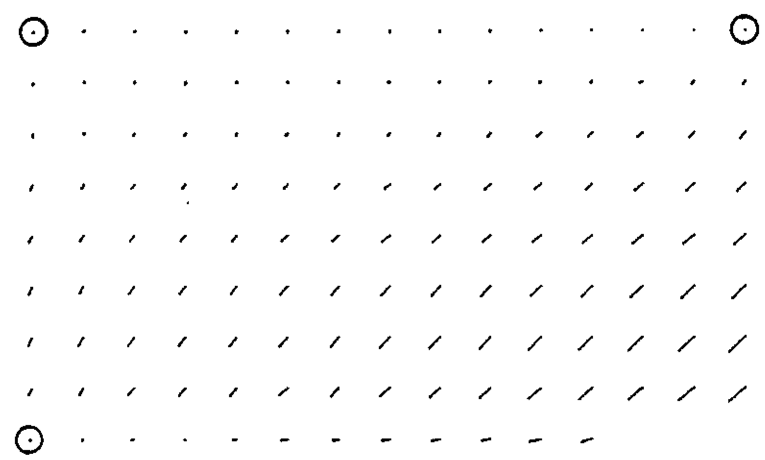


FIGURE 9 DISCREPANCIES WITH 3 CONTROL POINTS (CIRCLED) - rmse = 0.7mm

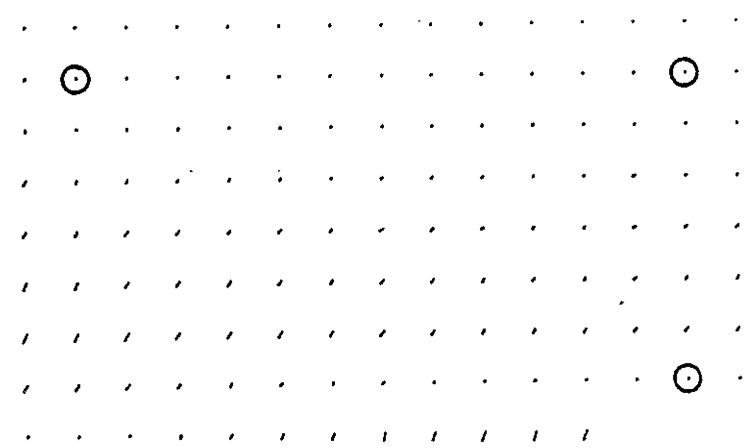


FIGURE 10 DISCREPANCIES WITH 4 CONTROL POINTS (CIRCLED)
- rmse = 0.3mm

FIGURE 11 DISCREPANCIES WITH 5 CONTROL POINTS (CIRCLED)
- rmse = 0.4mm

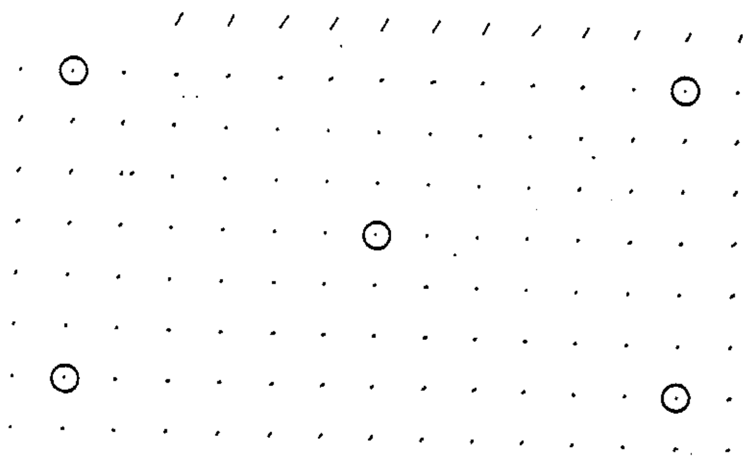


FIGURE 12 DISCREPANCIES WITH 9 CONTROL POINTS AND A PATCHWISE ADJUSTMENT
- rmse = 0.21mm

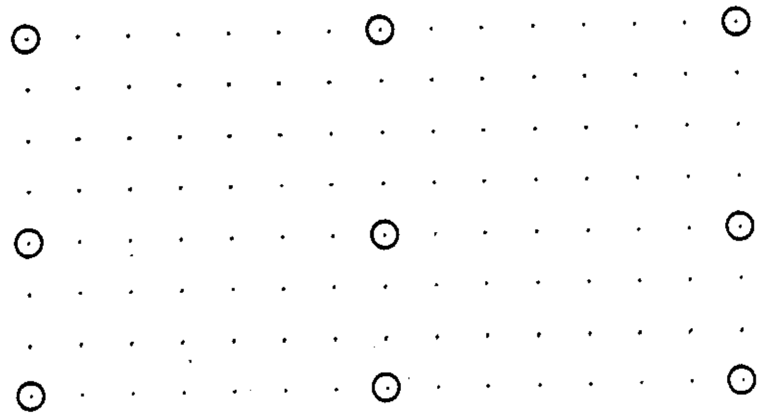
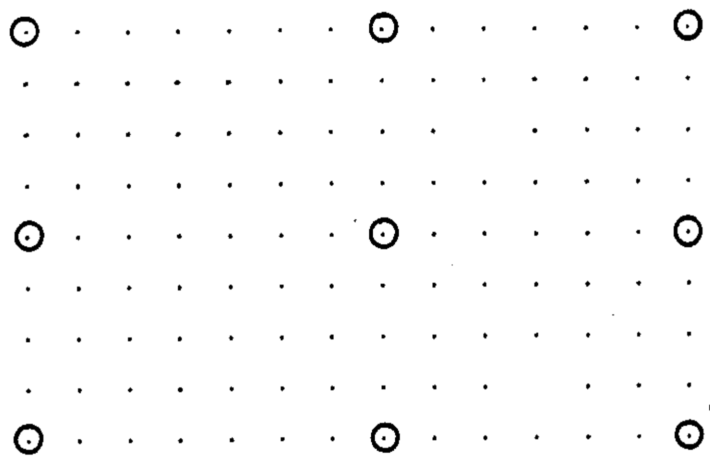


FIGURE 13 DISCREPANCIES WITH 9 CONTROL POINTS AND A PATCHWISE ADJUSTMENT
- rmse = 0.21mm



There are other ways to decide what is acceptable error in digitizing. The stated accuracy (presumably rmse) of some manual linefollowing cartographic digitizers is between 0.05mm and 0.25mm, although many manufacturers do not provide this statistic. The 'gut' reaction of mapmakers appears to be that any accuracy (rmse) better than about 0.20mm is alright! Standards published in 1986 [BECK and OLSEN, 1986] required for the United States that the rmse (with respect to the original map document) of digitizing be 0.13mm. This implies that the Scan-Cad attachment we tested would not meet the US standards (i.e. those required by the United States Federal Interagency Coordinating Committee on Digital Cartography).

In the 1970's the Topographic Service of the Netherlands (TDN) recorded the production positive positions of over 500 well defined details appearing on topographic maps. These recorded positions were compared with the positions of the same well defined features determined during photogrammetric control densification. It was assumed that the photogrammetrically derived positions were 'true'. This enabled determination of the accuracy of the cartographic processes (including photogrammetric plotting, cartographic drafting, and production positive preparation) found in mapmaking. The statistic created from this exercise was rmse. The value of rmse found in this exercise (from the 500 points) was 0.19mm [KERS, 1988]. Such an error in the map base was considered acceptable to TDN - they hoped to achieve an rmse of less than 0.20mm. In 'pre-digital days' maps represented the information base, but to extract information from this base required measurement. The precision of the measurement tools used ranged from 0.05 (a x10 linen measure) to 0.5mm (a ruler). Thus the accumulated error in the derived information was in the range 0.19 to 0.53mm. Digitizing replaces the act of measurement in digital information systems. Assuming the same maps are the information base, then digitizing represents a complete measurement exercise. Any dimensions subsequently required will merely result from the interrogation of the coordinates stored in the digitizing process plus the extremely small errors resulting from the mathematical processing of the computer. Following this reasoning as long as digitizing errors stay within the range of those normally found in manual map measuring then the digitizing errors are acceptable Adopting this more general approach to map information the Scan-Cad attachment is acceptable.

B2.2.3 TEST OF THE SCAN-CAD ATTACHMENT BY DTM GENERATION

A test has been performed to determine the acceptability of the Scan-Cad for capture of contours for use in Digital Terrain Model (DTM) creation [CHUNG, 1989]. This was a test comparing the results from scanning using the Scan-Cad attachment and the KartoScan scanner of SysScan.

One can predict two types of problems with a scanner of the Scan-Cad type, and these are related to its relatively low resolution and geometric instability - especially when comparing it to a high-end product such as KartoScan. These are:

1. the relatively low resolution of the Scan-Cad scanner may present problems for vectorising - as a result of clustering of the pixels from fine detail.
2. the relative instability of the Scan-Cad scanner may lead to positional errors in the captured data.

These two problems were addressed in the test. A contour separate was scanned by Scan-Cad scanner and also on a KartoScan scanner. The data from Scan-Cad was reformatted so

that it could be handled by SysScan's vectorizing software, and then the two sets of data were both vectorised following the same steps using SysScan software.

FIGURE 14 a,b,c,d shows the result of these two sets of scanning and vectorizing. Many more bridges remain in the data originating from the Scan-Cad than from the KartoScan. It took an inexperienced operator 5 hours to clear these bridges. This extra editing time is directly attributable to the lower resolution of the Scan-Cad.

Once the two sets of data were prepared they were used as input to generate two DTM's. Exactly the same techniques were used to generate the two DTM's. One can check height values for check locations given in a DTM against height values for the same places obtained by measuring a photogrammteric stereomodel. Differences between the DTM height values and the photogrammetric height values give an indication of the quality of the DTM. These differences can be used to obtain RMSE. This RMSE is also an indication of the geometric quality of the digital data representing the contours, which is in turn an indicator of the stability of the scanners.

The following results were obtained:

Rmse for a 50m DTM when source data is from Kartoscan	5.08m
Rmse for a 50m DTM when source data is from Scan-Cad	5.48m
Rmse for a 100m DTM when source data is from Kartoscan	6.55m
Rmse for a 100m DTM when source data is from Scan-Cad	6.55m
Rmse for a 150m DTM when source data is from Kartoscan	10.35m
Rmse for a 150m DTM when source data is from Scan-Cad	9.58m
Rmse for a 100m DTM when source data is from Kartoscan	10.87m
Rmse for a 100m DTM when source data is from Scan-Cad	10.57m

From these it can be seen that there is little difference between geometric quality of the products of the two scanners. Low resolution and the resulting edit times seem to be the main problem of the ScanCad.

FIGURE 14a CONTOUR SEPARATE SCANNED BY SCAN-CAD

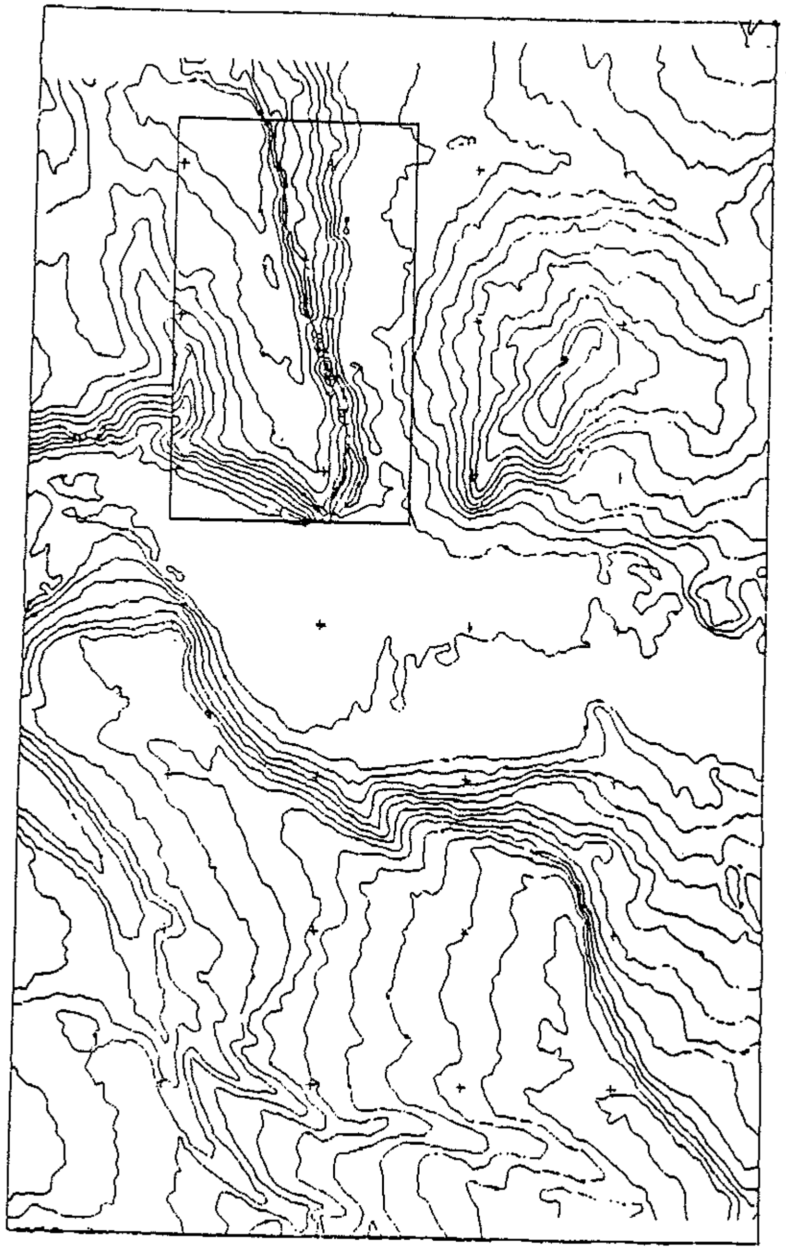


FIGURE 14b DETAIL OF A PART OF A CONTOUR SEPARATE SCANNED BY SCAN-CAD

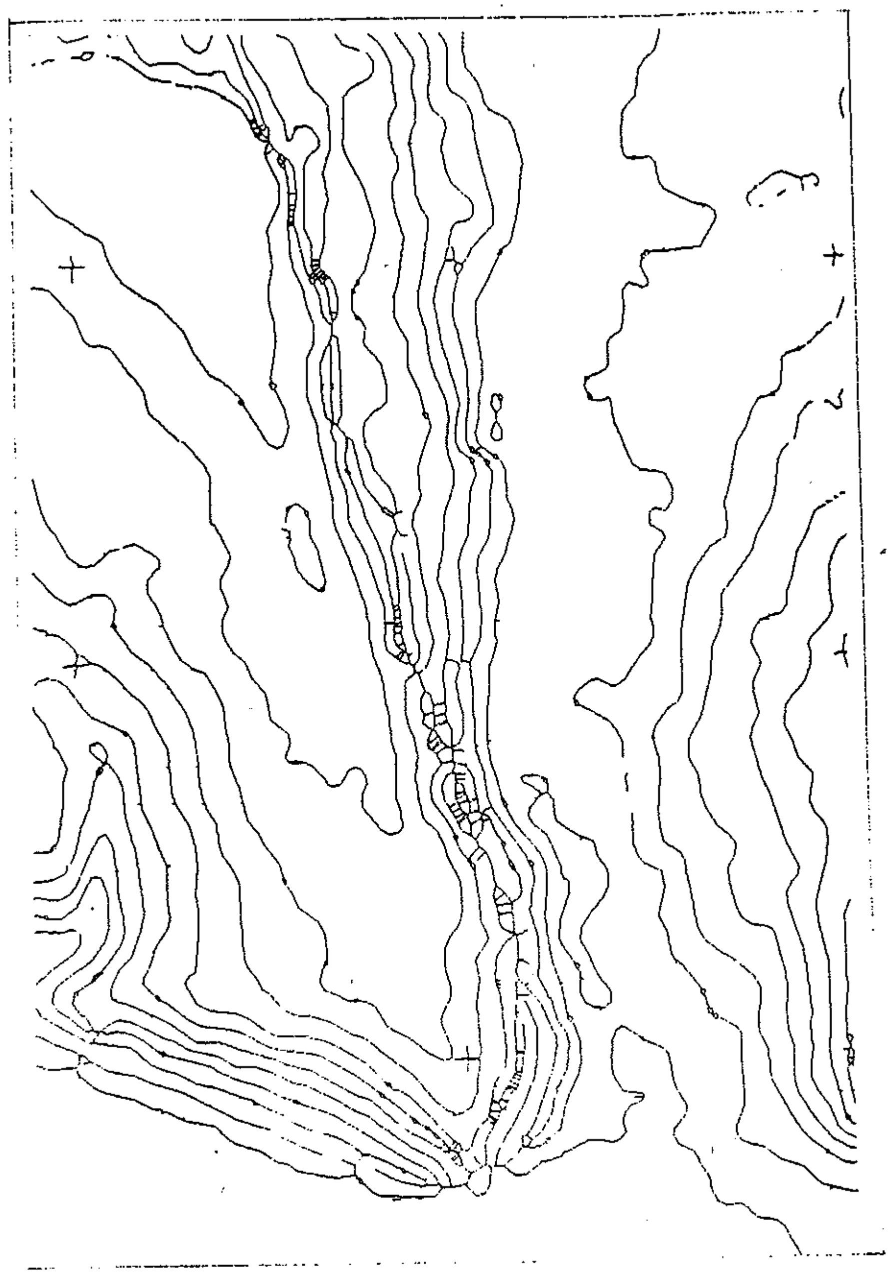


FIGURE 14c CONTOUR SEPARATE SCANNED BY KARTOSCAN.

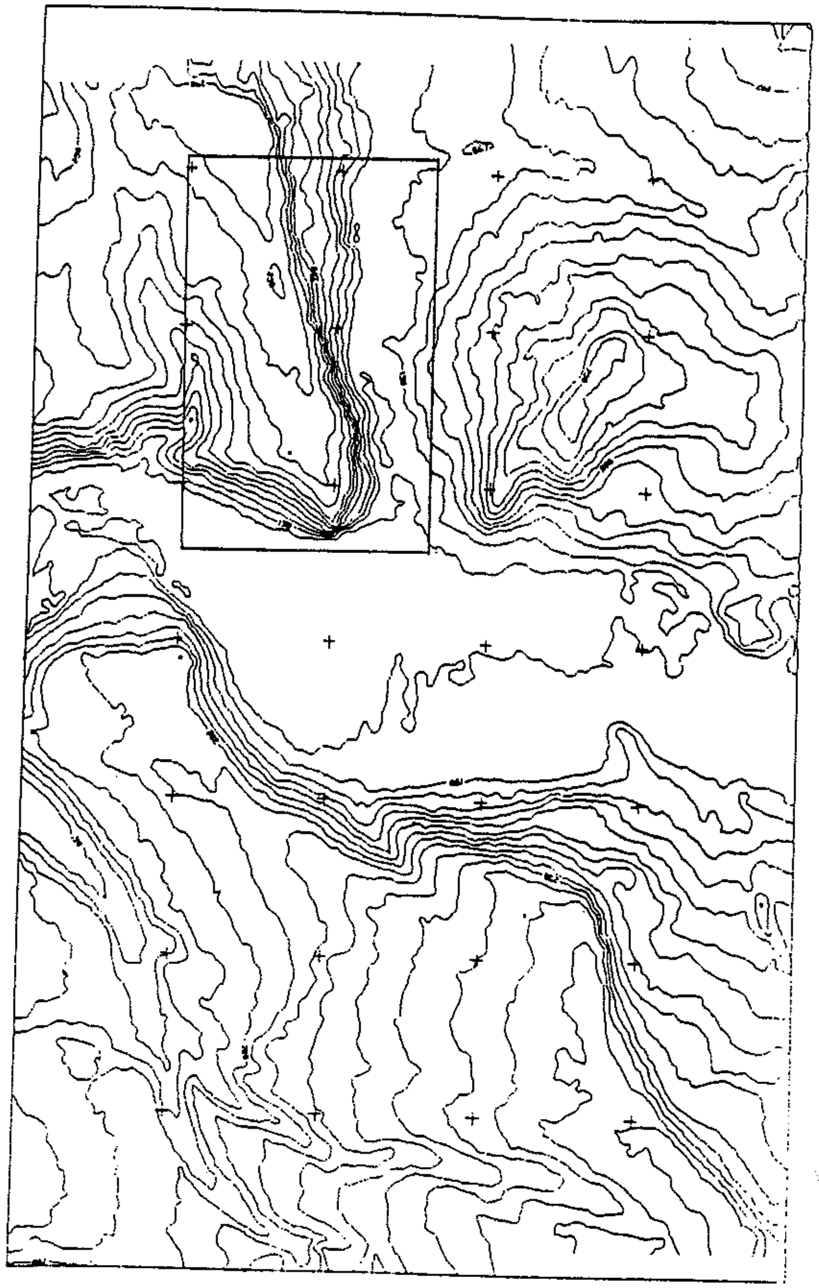
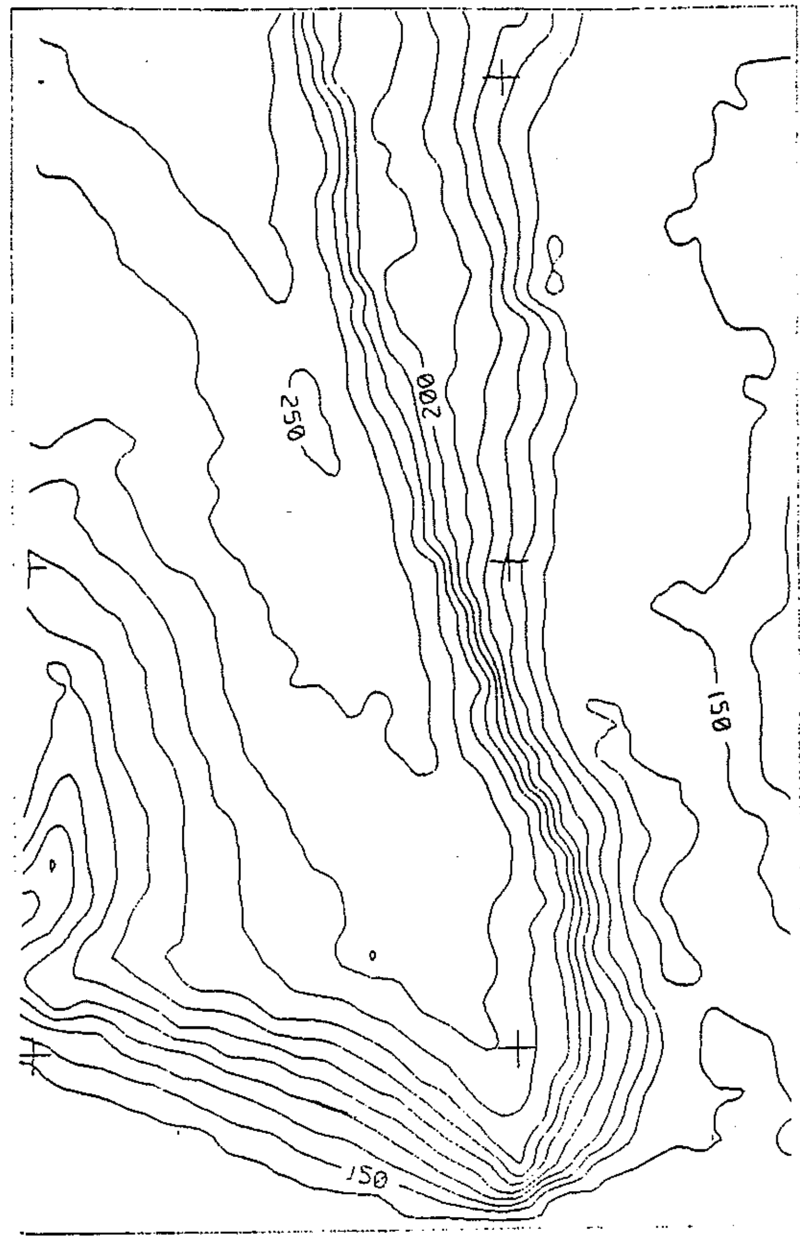


FIGURE 14d DETAIL OF A PART OF A CONTOUR SEPARATE SCANNED BY KARTOSCAN.



B2.2.4 THE AGFA FOCUS S800GS

The second scanner we tested is the new Agfa Focus S800GS scanner. An Apple Macintosh (SE or II) is required to control this scanner. The scanner is connected to the Macintosh with a SCSI interface. Interfaces for IBM will be available soon.

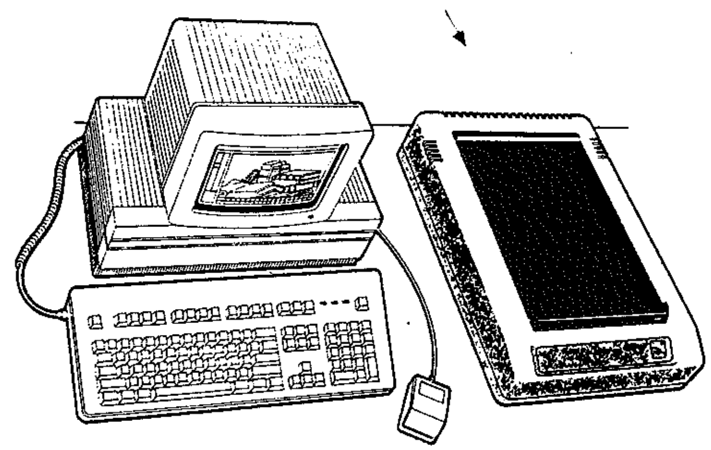
Of this scanner four different models are produced. Two models scan with resolutions of up to 600 dpi., and two models with resolutions of up to 800 dpi. Two of them can detect greylevels. The model names are Agfa Focus S800GS, Agfa Focus S600GS, Agfa Focus S800 and Agfa Focus S600 respectively. The model we tested was the Agfa Focus S800GS. Figure 15 shows this scanner.

The Agfa Focus S800GS is a flatbed scanner. The original document must be placed on a glass-plate. The maximum size that can be scanned is A4. Flexibility of the original is not a requirement, even a thick book can be placed on the glass-plate and its pages can be scanned. Alignment of the original is difficult. The McView Plus software allows for rotation of the scanned image. In our test we did not use this software feature because we did not want the accuracy of this rotation-procedure to interfere with the accuracy of the scanner itself.

Under the glassplate a lamp and a 21.6 cm wide strip of CCD's are positioned. An equally wide mirror is moved 29.7 cm along the long side of the original and reflects the image towards the CCD's. As the mirror can move in accurate steps that are as small as 1/800 of an inch the highest possible resolution in the y-direction is 800 dpi. The CCD's are knitted together as tightly as 400 per inch. By means of interpolation this 400 dpi resolution in the x-direction is increased to 800 dpi. The Agfa Focus S800/S800GS can scan in the following 13 sub-resolutions: 800, 700, 600, 500, 400, 350, 300, 250, 200, 175, 150, 125, and 100 dpi. The Agfa Focus S600/S600GS can scan in the following 13 sub-resolutions: 600, 525, 450, 375, 300, 263, 225, 188, 150, 131, 111, 94, and 75 dpi.

A word on the McView Plus software that comes with the Agfa scanners: McView Plus is a user-friendly program that can be used in three ways. The first application is its capability to control the scanner. After a proof-scan at low resolution is made, the user can select the part of the image to be scanned at the specified resolution. Originals with continuous tones can be scanned with 64 levels of grey. The user can choose to store all real greyvalues in a raster file in TIFF format. He can also choose to store the image as a screened halftone image. The screening is applied during the scanning-proces. The second application is raster editing. The raster file can be changed using an "eraser', a "pencil", fill patterns, text can be added, etc. After scanning and editing McView can save the raster files in a number of file formats. Many raster file formats can be imported by McView as well, so McView can be a very useful program for raster-file conversion. The third application of McView is vectorisation of features from raster files. (This third part of McView does not work when there is no Agfa Focus scanner connected to your Macintosh.) McView offers the choice between vectorizing of 'outlines' and vectorising of 'skeleton lines'. When 'outlines' is chosen, the program draws vectors around all black area's. If the original raster file contains greyvalues the program draws vectors on the edges of areas of uniform greylevel. (The number of greylevels can be specified by the user.) When 'skeleton lines' is chosen the program draws the vectors of the centre lines of all line features.

FIGURE 15. THE AGFA FOCUS S800GS SCANNER



B2.2.5 GEOMETRIC TEST OF THE AGFA SCANNER WITH A TEST GRID

An almost identical approach was used with testing the Agfa Focus S800GS Scanner as was used with Scan-Cad. This was that a set of crosses produced on a very stable photo-optical plotter was used as the test document. The crosses were exactly the same in dimension and lineweight, but because of the smaller format of the Agfa scanner and the resulting need to cut up the test document for scanning, the crosses were positioned every 2.5cm.

The sheet had to be cut into eight pieces - each about A4 in size, and all were scanned at 127micron resolution. This resolution was selected as it was the same as available on the Scan-Cad. Higher resolutions are possible on the Agfa scanner. The scanned data (in .EPS format) was transformed into the .LSC format of the SysScan system for subsequent vectorizing. The vectorised data was then processed in the same way as that which had been derived from the Scan-Cad attachment. Four control points were used. TABLE 3 gives the accuracy findings for the four corner pieces of the sheet.

TABLE 3. ACCURACY FINDINGS OF THE AGFA FOCUS S800GS SCANNER

	RMSE	MAXIMUM ERROR
Test Document Piece 1	0.14	0.36
Test Document Piece 4	0.11	0.33
Test Document Piece 5	0.12	0.34
Test Document Piece 8-1	0.14	0.35
Test Document Piece 8-2	0.12	0.40
Test Document Piece 8-3	0.12	0.32
Test Document Piece 8-4	0.13	0.36

It can be seen from TABLE 3 that this device gives values comparable to a more expensive scanner. To test its stability Piece 8 was scanned four times. The derived coordinates at five check crosses (which were not control points) are given in TABLE 4.

TABLE 4 DERIVED COORDINATES AT FIVE CHECK LOCATIONS

SCAN NR. X,Y(mm)	1	2	3	4	SDx,SDy
PNT 1	99.89,100.33	99.77,100.34	99.77,100.23	99.88,100.34	.06,.04
PNT 11	150.03,125.05	149.95,125.05	149.97,124.98	150.03,125.05	.03,.02
PNT 39	250.06,199.87	250.06,199.87	250.06,199.87	250.06,199.87	.00,.00
PNT 78	225.10,324.95	225.10,324.93	225.10,324.93	225.10,324.93	.00,.01
PNT 88	275.09,349.92	275.09,349.94	275.09,349.94	275.09,349.94	.00,.01

It can be concluded that the standard deviation (SD) in derived coordinates at each check point indicates that the Agfa scanner as a digitizer is as precise as is required for Cartography, although there is evidence of slightly less precision along one side of the scanner.

FIGURE 16 shows the vectorised crosses and a discrepancy diagram for one of test pieces.

FIGURE 16a. AGFA SCANNED TEST CROSSES AFTER VECTORIZING

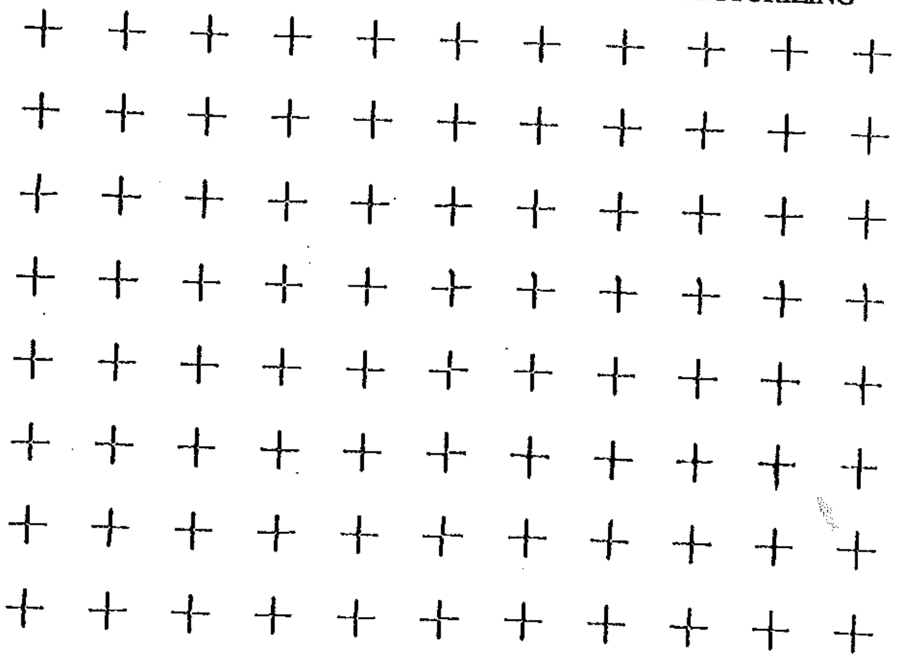


FIGURE 16b. AGFA SCAN TEST DISCREPANCIES (arms at scale 1:1)

B3. PHASE I CONCLUSIONS

Low-cost scanners are constrained by their instability, resolution, or format. Accepting the rule of thumb that the stated resolution should be half the minimum linewidth on the scanned document, then only a few of the low-cost scanners (having resolutions better than 0.50mm) could pick up fine map detail. The Scan-Cad attachment is a convenient device as it scans large format maps and uses as a drive a device which may also be used as a plotter. At this stage it seems that a five centimeter strip along the top and bottom edge of the scanner should be avoided, but that otherwise a 0.2mm rmse with a maximum error of about 0.5mm can be expected - although further confirmation of this is required. Further tests should be carried out to check the stability of the Scan-Cad attachment, and the usefulness of smaller and possibly more stable scanners should be examined.

The Agfa S800GS Scanner has a resolution of 63 microns. It was noticeable good at picking up detail, and its geometric precision (RMSE 0.1 mm) is high. Its main disadvantage is its A4 format.

Consideration should be given to the role of a low-cost scanner with rmse of about 0.2mm in mapmaking, such as in DEM generation and mapping features whose boundaries are known to be imprecisely defined (e.g. soils). But, it would be interesting to test a device like Scan-Cad having the higher resolution of the Agfa scanner.

C. PHASES II AND III OF THE STUDY EXAMINATION OF SOME PROBLEMS IN AND A PROPOSAL OF SOME SOLUTIONS TO VECTORIZATION AND AUTOMATIC FEATURE CODING

C1 THE RESEARCH STRUCTURE

In section A4.3 problems, associated with scanning and processing scanned data, as indicated by our third questionnaire, were outlined. In this section of our report we will consider specifically two aspects of these problems, namely:

vectorization, and
automatic feature coding.

These two 'main' problems were, according to the answers of the third questionnaire, further broken down as follows:

A. PROBLEMS ASSOCIATED WITH VECTORIZING & RELATED BATCH PROCESSING

In general these are:

loss of detail;
distortion of detail; and,
introduction of artefacts.

Specific questions raised within these three groups are:

1. Loss of Detail

1.1 How can we decide the resolution at which a map should be scanned, taking into account e.g. linewidth, gap width, scanning times, interactive editing times, etc.?

1.2 How can automatic cleaning of the raster data set be performed without loss of useful information?

1.3 How do we cope with contour clustering, assuming an otherwise optimal resolution is used?

2. Distortion of Detail

2.1 How serious, in metric terms, is intersection node displacement when a thick and thin line intersect? How does this arise? If this is a serious problem, how can it be overcome?

2.2 How serious, in metric terms, is intersection node displacement when one or both of the intersecting lines is curved? How does this arise? If this is a serious problem, how can it be overcome?

2.3 How do we re-establish the correct corner angle when vectorizing has distorted it?

2.4 How can it be ensured that the correct dangling node is found?

3. Introduction of Artefacts

3.1 How can it be ensured that bridges do not form or are successfully removed at angled intersections.

3.2 How can it be ensured that bridges do not form or are successfully removed at locations containing fine detail, such as close parallel lines, text, point symbols and point symbols incorporated into patterned lines?

3.3 How can artefacts be prevented from being generated when lines of uneven thickness are vectorised?

3.4 How can it be ensured that bridges do not form when a line is highly tortuous?

B. PROBLEMS ASSOCIATED WITH AUTOMATIC FEATURE CODING (OR THE SEMANTIC ASPECTS OF AUTOMATIC DIGITIZING)

In general these are:

Recognition of Lines;
Recognition of Point Symbols.

Specific problems within these groups are:

4. Recognition of Lines

4.1 How can we ensure that any line made from pecks, point symbols etc. can always be recognised as a line in spite of gap sizes elsewhere on the map and the tortuousity of the line itself?

4.2 How can we ensure that lines crossing a patterned background (screens, hatching, hillshading) are recognised?

4.3 How can we ensure that line continuation is found despite gaps such as those formed by deleted contour values or the partial elimination of contours in areas of steep terrain?

4.4 How can we ensure that feature codes based on linestyle, weight, colour, pattern, or context can be attached to lines?

5. Recognition of Point Symbols

5.1 How can we ensure that simple geometric features such as circles, arcs, crosses, text, utilities' symbols etc. are correctly recognised and identified, even when they are close to line symbols or each other, oriented at different angles, and of varying size?

5.2 Which pattern recognition tasks are best performed when the data are still in raster form and which benefit from the data being in vector form?

It had been hoped to address each of these questions individually, and find solutions to them, but it became obvious, in our reading, that all questions were interrelated.

Instead we decided to approach this phase of our work by first identifying the processing steps between scanning and acquiring clean vectorized and feature coded data. Once these processing steps had been identified it became possible to see whether they represented:

1. a solved problem;
2. a problem with a variety of solutions of varying quality; or,
3. a problem with no reported solutions.

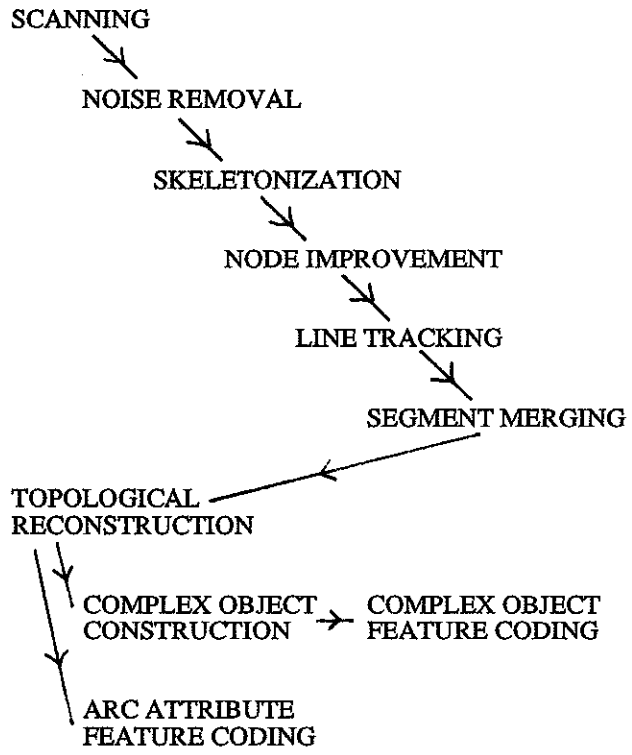
By adopting this procedural approach, rather than the goal directed approach indicated by the list of questions 1.1 to 5.2 we thought it made the subdivision of research tasks more easy, and prevented overlap in our work. Obviously on completion of our work it becomes necessary to return to the questions 1.1 to 5.2 to see if we have answered them!

After considerable discussion, it was decided that the procedural steps taking us from the scanned document to the clean, vectorized, and feature coded data are those shown in FIGURE 17.

Another approach would have been to deal with the problems pertaining to particular map scales. Small and medium scale maps are coloured and have much more complex symbology than larger scale maps, so entirely different problems may arise in their automatic digitizing. However one might alternatively suggest that the differences are merely scale and emphasis (e.g. there is more area symbolisation in smaller scale maps than large), and solutions for large scale maps will also work on small scale maps. We decided not to distinguish between solutions based on scale.

A classical approach to research is to start with a literature review. As our approach was to directly address the problems identified in the questionnaires, there was no need to search the literature to re-identify these problems. Instead we took proposed solutions (found in the literature) which claimed to solve the problems and determined whether they did or not. If they did not we tried to improve them. As far as literature is concerned, we have created a computerised literature database, including summaries of the references, some details of this are in APPENDIX F.

FIGURE 17. SCAN DATA PROCESSING STEPS



This set of procedures indicates that the goal we really seek is a set of feature-coded ARCS and COMPLEX OBJECTS. There is a need to discuss and clarify this, as it greatly influences the structure of our research - the topic of this section.

There are several ways to organise the information contained inside a geographic database, including pixels, points, lines, and areas. Implicit behind all of this Working Group's activity is the assumption that although pixels are gathered during scanning they are an inadequate structure for the purposes of topographic or cadastral map production. Few map makers would disagree that the objects stored in a data base must represent the real world entities which have been traditionally recorded on maps as:

point features;
line features; and,
area features.

But, there is no reason why geographic entities have to be stored in a GIS in the same way they have been, for hundreds of years, on maps. Even if the GIS is to be used primarily for map production, certain features only implied in a traditional map must be explicitly recorded in the database of a GIS - especially if this is useful in some cartographic tasks, such as generalisation or area patterning.

One implied feature is a network. On topographic maps the following networks are found:

roads;
railways;
utilities (water, gas, electricity etc.);
hydrography, etc.

Once in a GIS, network information can be used for generalisation (identifying the main stream of a river network, deleting headwater streams), route finding (shortest or alternative routes between locations), etc. But what information do we need about a network in our GIS? To answer this we must realise a network is characterised by:

paths connecting junctions (or places);
paths ending; and,
junctions at which it is possible to follow 3 or more different paths.

The objects involved in this characterisation are:

arcs or 'edges' (path concept); and,
nodes or 'vertices' (junction and end concept).

If the user of a GIS requires topographic maps he is requiring a graphic representation of the world following a certain geographic model. Although the model really bears little relationship to life in the late twentieth century - it is closely linked to military life in the nineteenth century and the problems of moving armies around - it is a model which is widely applied. This model is that the world is composed of a patchwork of land-cover polygons held together by a framework formed by the communications network. Superimposed on this patchwork and it framework are some other independent networks - such as the railways and utilities. (There are also independent point objects which are scale dependent such as trees, churches, and villages and we may consider them to be complex objects. Their independence means that we can evade the responsibility of considering their topology.) So, for the purpose of map production land-cover polygon information is very interesting; adding a third class of objects - the domains- provides the user with this opportunity.

A domain is defined by its boundary - conisting of node-linked arcs. The boundary is shared by n domains. So we can consider storing arc, node, and domain information in a GIS. Examples of such data structures which have been implemented in GIS's are DIME and POLYVRT.

The data structure (arcs, nodes, domains) may be stored in 4 major tables:

1) arc topology table;
2) domain topology table;
3) node geometry table; and,
4) arc geometry.

These tables are summarised in APPENDIX E.

Of course, as well as having the arc topology, domain topology, node geometry, and arc geometry tables, there will be tables containing attributes of the database objects listed in the geometry and topology tables, and other intermediate tables as indicated in sections C5.5, C5.6, C6.2.1. Automatic digitizing must help as much as possible to determine the contents of these tables.

At the beginning of this section we said that our goal is a set of feature coded arcs and complex objects and the procedures to achieve this are listed in FIGURE 17, but, it is possible that other procedures become significant if other goals are sought. To this end it was decided to present a discussion of 'Other Approaches' also.

In this part (PART C) of our report each of the procedures described in FIGURE 17 will be addressed, but before concluding the report, other approaches will also be considered, briefly (Section C9.).

C2. SCANNING

Scanners record, pixel by pixel, a value representing the contents of the document being scanned. Scanning hardware can now be considered reliable, and options available to the user consist of:

1. format;
2. resolution (pixel size);
3. value recorded (e.g. grey level, black/white, colour); and,
4. an ability to calibrate the scanner so it records classified values.

It will be shown in this report that some fundamental processing problems remain in the handling of scanned map data (e.g. node displacement, section C5). As a result workers may have to reconsider their options with regard to the scanning itself.

Alternatively, if problems do appear insurmountable, workers may question the need to vectorize everything.

We will return to these options.

C3. NOISE REMOVAL

Maps which are to be scanned always contain unwanted marks (e.g. optical noise and dirt) which remain as noise in the scanned data set. It may be advantageous to attempt to remove this noise (often called 'salt and pepper' noise) before skeletonization. Also noise may be removed at other stages during processing, such as after skeletonization, or even vectorization. Noise removal at various points in processing will be considered in the following sections.

C3.1 COMPONENT LABELLING

If several pixels are connected it becomes evident to the human observer that they belong to the same entity.

An entity, for the purpose of this section, is either:

an object; or,
background area.

All entities are called connected components and an image can be described in terms of its connected components. The pixels of either an object or background area are connected to other object or background area pixels respectively. However it has been suggested that background area pixels connect to other background area pixels in a way that is different from that with which object pixels connect to other object pixels, therefore an analysis of the connectivity of a pixel will indicate whether it really is an object pixel.

Two types of connectivity can be chosen:

4-connectedness where connected pixels are direct neighbours of each other;
and, 8-connectedness where connected pixels are indirect and direct neighbours of each other

The following diagram clarifies this:

```
0 0 0 0 0
0 0 1 0 0
0 1 0 1 0
0 0 1 0 0
0 0 0 0 0
```

All 1-pixels have 8-connectedness with other 1-pixels, but do not have 4-connectedness with any other 1-pixels. The central pixel (an 0-pixel) has only 8-connectedness with other 1-pixels, but not 4-connectedness.

It has been suggested [DEUTSCH, 1972] that the component pixels of black objects on a white background can be identified using 8-connectivity for objects and 4-connectivity for background components. That is, that a black pixel truly belonging to an object can be 8-connected to the other object pixels, whereas a white pixel which is truly a background pixel is 4-connected to background pixels.

A useful tool is connected component labelling. It is described by Rosenfeld [ROSENFELD et al., 1976], and can be used to produce a map of labels with a different label for each object, but the same label for each pixel belonging to an object. The connectivity analysis briefly described in the preceding paragraph can be used to identify and label the connected components.

Another approach to connected component labelling has been developed using 4-connectivity for background components. Its output is another map showing the different background components labelled according to the 4-connectivity.

Such labels as in the two abovementioned maps can be used to count objects ('How many objects are in the picture ?') and to make area measurements [ROSENFELD et al., 1976]. The usefulness of component labelling to our task is discussed in the next section (C3.1.1, C3.1.2, and C3.1.3).

C3.1.1 AN APPLICATION OF COMPONENT LABELLING

In this project, we have applied component labelling twice: in pre-processing noise removal (described below in sections C3.1.2 and C3.1.3) and in topological reconstruction (section C8.2.3).

One must remember that component labelling refers to the idea of region. An object is a set of pixels which can always be described as a surface or region; even a single pixel can always be perceived not as a point but as a square surface.

C3.1.2 NOISE REMOVAL BASED ON AREA DIMENSIONS

It can be assumed that an object whose area is small is noise. (If this noise is isolated from non-noise objects it is often called 'salt and pepper' noise.) A threshold minimum area, above which an object is not noise, can be assumed. Component labelling can be used to determine the area (expressed in the sum of pixels) of an object's surface.

Two different thresholds can be used - one for objects and one for background. In practice, these thresholds will be set to values depending on the size of noisy dots (small ink dots and scanning noise) and therefore on the scanning resolution.

C3.1.3 NOISE REMOVAL BASED ON PERIMETER DIMENSIONS

Another component labelling noise removal technique was then tackled in which deletion of an object occurred when its contour (edge) had a negligible perimeter. If this technique is applied, contour (edge) pixels have to be identified at an earlier stage in the processing. As with all pixels in a component (object), contour (edge) pixels are also component labelled. The number of contour (edge) pixels belonging to each object can be summed - thus generating the object's perimeter. It is then possible to build a relation (a database table) indicating the perimeter of each object. Objects with a perimeter less than a certain minimum length can be deleted as noise.

C3.2. NOISE REMOVAL AT SEVERAL STAGES DURING PROCESSING

Methods so far described have dealt with the deletion of salt and pepper noise. The methods used to detect noise are mainly those of size.

Here we present a method of noise deletion which deals with salt and pepper noise and contour (edge) noise. It processes the image from its original raster form and applies shape criteria.

It also should be realised that this method makes use of temporarily generated vectors, so, with a few adaptations, it can also smooth a vector image. However, these adaptations will not be further elaborated here.

C3.2.1 NOISE ON MAPS

Most graphic entities on maps are built of graphic primitives which are smooth and sharp. Arcs traverse the map in a smooth way. When they perform an abrupt change in direction, we know the arc ends and a new graphic primitive starts. Furthermore, the arcs are drawn sharply, that is the zone between the arc and the neighbouring feature can be defined precisely. We can say that graphic entities, which do not satisfy these criteria are distorted with noise.

C3.2.2 DETECTION OF NOISE

When we look at a map arc's raster representation we expect to find a contour (edge) consisting of pixels whose Freeman codes (see APPENDIX D for a description of Freeman codes) show a more or less gradual rate of change. When this rate of change varies considerably over a short part of the contour (edge) we have either detected noise or an arc end. Also we can say that an arc is characterized by a sequence of Freeman codes consisting of two gradually changing sequences separated by an abrupt change. Thus we can distinguish noise by examining configurations of the Freeman codes of contour (edge) pixels. However, as Freeman codes only give local information, we need to employ sequences of considerable length to detect gradual rates of change, requiring much computation time - and this is why we suggest that it is better to employ vectors instead of Freeman codes.

We will now set up criteria to distinguish noise from the rest of the image. From the preceding paragraph we can conclude that noise only occurs where a non-gradual change in contour (edge) direction appears. We also can say that the magnitude of vectors describing the noise on a contour (edge) will be small and that the summed length of these vectors is also small. Furthermore, the absence of continuity implies that the noise contour (edge) will 'double back' on itself. That means that the ratio between the direct distance between two pixels of the noise contour (edge) and the distance along the noise contour (edge) between these pixels, is small. Thus, the vectors describing the noise contour (edge) have a start point and an end point, which satisfy this small ratio.

Summarizing we can say that a certain sequence of (n) vectors, with lengths l1, l2..li..ln, startpoints s1, s2..si..sn and endpoints e1, e2..ei..en, describing part of the contour (edge) of an object can be characterized as describing a noise element on an object if it satisfies the following criteria:

1) $li < lmax$;

2) $d(s1,en) / \sum_{i=1}^{n} li \leq R$;

3) $n < I$;

where I and R represent thresholds.

We have been dealing only with contour (edge) noise. It is clear however, that salt and pepper noise represents a special case of contour (edge) noise; the case where R in the above condition (2) equals zero. Thus the method is also capable of detecting salt and pepper noise.

C3.2.2.1 DESCRIPTION OF AN ALGORITHM FOR REMOVING NOISE VECTORS

The algorithm, introduced in the preceding section, starts with vectorizing the contour (edge). Then, starting at the first vector of a chain, it searches until a long vector is found. This vector is stored as the 'end vector', so when the whole contour (edge) is processed it knows to terminate processing at the 'end vector'.

After this the contour (edge) is followed further until a short vector, the 'start vector' (giving us a start point), is found. This start vector is also stored.

The algorithm then continues to follow the vectors of the contour (edge) until another long vector is found, in the meanwhile keeping track of the summed length of the followed vectors.

The algorithm then computes the ratio outlined in equation (2) above. If this ratio exceeds the threshold the algorithm then steps back one vector and repeats the calculation, continuing to do this until it arrives at the immediate successor of the 'start vector', or until the ratio is smaller than R.

If it arrives at this immediate successor it stores its vector as a new 'start vector' and repeats the whole process again.

Otherwise when, at a certain vector (the 'stop vector') the ratio is smaller than the threshold and different from zero (so it is not salt and pepper noise), it may be assumed noise has been found between the 'start vector' and the 'stop vector', but, a final test is made, the reason for which is explained four paragraphs below.

An improvement to the above is needed for contours having a gradual curve at all places. While following this contour we will always arrive at the vector we started from since the condition for calculating the ratio is never met. As the ratio at this place is 0.0 all vectors are deleted. The object is considered to be salt and pepper noise, even though we might have traversed the whole map while following the contour. On such a curve all vectors are short but not noise. So instead of looking for a long vector after the 'start vector', the nth. vector after the 'start vector' is treated as a long vector, and processed as already explained.

When the difference in direction between the predecessor of the 'start vector' and the successor of the 'stop vector' is considerable the existing intermediate vectors are left unchanged; otherwise the 'start vector' and the 'stop vector', together with their intermediate vectors are replaced in the chain by one vector connecting the startpoint of the 'start vector' and the endpoint of the 'stop vector'.

Then starting from the successor of the newly created vector the contour (edge - see section C.4) is followed again until the next small vector is found, which subsequently is stored as a 'start vector', and the whole process is repeated. This goes on until the 'end' vector is met for the second time.

The reason for the final test (mentioned four paragraphs above) is to distinguish between junctions and noise. Without applying it there is no difference between noise and a junction, because a junction will have small vectors immediately before and after it. When we apply the final test we are quite sure that the contour (edge) segment we are processing does not contain a junction, because the contour (edge) goes on in more or less the same the direction, after the noise as before the noise.

After the process has terminated, the only information which is left, is a chain of vectors describing the contour (edge) of an object. All information about individual pixels near the contour (edge) is lost. To get the raster representation back we propose the following method. We start with an image which is completely empty, that is which contains no object pixels. We then rasterize the vector chain, store the information in the image, and turn all the pixels inside the contour (edge) pixels to object pixels (such rasterization procedures are described in any Computer Graphics textbook).

A less verbal description of this algorithm for noise removal by vectors, having as a product a pointer chain of vectors, follows:

```
PROCEDURE NOISE_FILTERING;

INPUT : chain of contour pixels;
OUTPUT: image : matrix of pixel values.

    BEGIN
    vectorize_contour(image, chain of contour pixels);
    go to first_vector in chain;
    current_vector = first_vector;

    WHILE vector is small DO
        current_vector = next_vector;
        stop_vector = current_vector;

    REPEAT
        WHILE current_vector is big AND
            current_vector <> stop_vector DO
            current_vector = next_vector;

        IF current_vector = stop_vector goto exit;
            (comment: we are at the first small vector)
        store direction previous vector;
        initial_vector = current_vector;
        initial_vector = start_vector;
        i = 0;

        WHILE current_vector is small AND i < I DO
            BEGIN
            current_vector = next_vector;
            i = i + 1;
```



```
            if current_vector = initial_vector THEN
                delete_intermediate_vectors;
            (comment: preceding test finds salt and
                pepper noise)

            END;

        REPEAT

            calculate ratio;
            i = i - 1;
            current_vector=previous_vector
        UNTIL ratio < R OR i = 2;

        IF ratio < R THEN
            BEGIN
            compare predecessor's direction with successor's;
            IF directions are unequal goto exit;
            delete intermediate vectors;
            END;
        current_vector = next_vector;
        exit;
    UNTIL next_vector = stop_vector;

    delete image;
    rasterize_contour;

    fill_contour;
    END;
```

C3.2.2.2 EXPECTED LIKELIHOOD OF SUCCESS

The proposed method does not guarantee a 100% noise free image. It is probably never able to give such a result. However, it will reduce the noise in an image considerably. Tests on an artificially made image showed 30% decreases in the number of noise vectors. However, these were not real scanned images. On the other hand the method may have the result of distorting the image in such a way that the result is far worse than the original noisy image.

The likelihood of either result depends heavily on the chosen threshold; if large a lot of noise will be deleted, but the image also will be distorted; if small, then the image will not be distorted, but probably hardly any noise will be deleted.

Shortage of time prevented us from testing the method for different thresholds. We suggest that this should be part of future research.

C3.3 BRIDGING

If it is assumed that the original map is well produced, then bridging originates from scanning at too coarse a resolution. Considerin the optimal resolution at which scanning can take place, in section B2.1 of this report a rule of thumb is given suggesting that scanner resolution should be at least half the minimum line width appearing on the document to be scanned. However the problem of bridging arises because gaps between lines are not detected. Thus the rule of thumb should be reworded: the scanner resolution should be at least half the minimum detail width appearing on the document. Detail may be line or space. This rule of thumb is more formally supported by the Whittaker-Kotelnikov-Shannon theorem [ROSENFIELD, et al., Vol I,p64].

However increasing resolution increases the required storage space, so other solutions - such as specifically finding and dealing with bridging have to be sought.

In most cases bridging occurs when lines approach and separate at low angles. This means bridging may have occurred at junctions having a particular form. Finding junctions of particular forms is described in section C5.5; the forms described in that section would not be suitable for locating bridges, but a similar approach could be adopted.

Even if a node having bridging potential has been found, it still has to be determined whether bridging has actually occurred. A suitable test is indicated in the diagram below, white noise (isolated '.' pixels) is likely to exist where bridging arises. If the test is passed, thinning between the lines can take place.

```
xxxxx..............................................xxxxx
xxxxxxxxxx....................................xxxxxxxxxx
xxxxxxxxxxxxxxx..........................xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx................xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxx......xxxxxxxxxxxxxxxxxxxxxxxxx
.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.....
..........xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..........
Possible bridge.xxxxxxxxxxxxxxxxxxxxxxxxx...............
....................xxxxxxxxxx.xxxx.......white noise..
.small angles.......xxxxx.xx.xxx.xx.....................
...............xxxxxxxxxxxxxxxxxxxxxxxxx................
..........xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..........
.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.....
xxxxxxxxxxxxxxxxxxxxxx..........xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx....................xxxxxxxxxxxxxxxxxx
xxxxxxxxxxxx..............................xxxxxxxxxxxxx
xxxxxxx.........................................xxxxxxxx
xx...................................................xxx
```

C4. SKELETONIZATION

For the last thirty years the resolution of scanners has increased, and as a consequence the extraction of information has become more complex. If maps with lines on have been scanned so that they are only one pixel thick it is easy to extract the lines by tracking them, pixel by pixel. But, for this tracking, if the lines are more than one pixel thick they must be reduced to a minimal set of data representing the lines and satisfying the following criteria:

1. lines must be one pixel thick; and,
2. the new representation must retain the shape of the original line.

Frequently in mapping we are dealing with elongated objects, so the ideal solution is the same as that used in manual digitizing, which is to extract the centreline (or medial axis). This centreline representation of the pixels generated by linear objects on a map may be called the Medial Axis Transform. The procedure generating the Medial Axis Transform is Medial Axis Transformation, or skeletonization.

There are several skeletonization algorithms, but all are essentially iterative, with each iteration giving an object thinner than the input object and working on the basis of 'peeling off' the outer edge (or 'contour' as it is often called in computer science) of the object. Without any controls the object would eventually be completely peeled away, so special conditions must exist in an algorithm to stop when the object is thin enough.

C4.1 CLASSICAL SKELETONIZATION ALGORITHMS

In the following discussions the pixel members of a 3x3 matrix may have to be referred to individually, the following labels will be used:

```
P3 P2 P1

P4  X P0

P5 P6 P7
```

(The central pixel is not labelled). Pixels may also be referred to as neighbouring pixels, all the pixels P0 to P7 are neighbours of X, but P0, P2, P4, and P6 are direct neighbours while P1, P3, P5, and P7 are indirect neighbours.

Classical skeletonizing algorithms (e.g. [STEFANELLI et al., 1971]) are characterized by having the following common features:

* background pixels are not processed;
* object pixels consist of edge and interior pixels;
* objects are only skeletonized by removing edge pixels;
* a series of (n) TESTS distinguishes between (i) the skeletal pixels - which have to be retained and form final skeleton, and (ii) the pixels to be removed;
* these tests involve the use of 3x3 pixel matrices;
* the pixels to be removed are removed in parallel;
* pixel removal (thinning) takes place until the skeleton should not be thinned any more; and,
* the set of skeletal pixels forms the skeleton.

In the late 1960's parallel processing of the above procedure was first considered, and consisted of:

```
REPEAT
    FOR EACH PIXEL DO
        IN PARALLEL
        IF SKELETAL THEN FLAG AS SKELETAL
        ELSE DELETE
    UNTIL ALL PIXELS ARE SKELETAL
```

However when processing pixels in parallel it has been found that using a 3x3 window leads to (i) disconnected objects, or (ii) the total deletion of some objects, such as lines which are 2 pixels thick or another even number of pixels thick. This is demonstrated in the following diagrams:

a) the original object

```
0000000000000000000000
1110000000000000000111
1111111111111111111111
1111111111111111111111
1111110000000000000011
0000000000000000000000
```

b) deletion of non-skeletal edge pixels - first iteration

```
0000000000000000000000
...0000000000000000...
111................111
111111..............11
......00000000000000..
0000000000000000000000
```

c) object after the first iteration - a disconnection has arisen

```
0000000000000000000000
0000000000000000000000
1110000000000000000111
1111110000000000000011
0000000000000000000000
0000000000000000000000
```

d) deletion of non-skeletal edge pixels - second iteration

```
0000000000000000000000
0000000000000000000000
...00000000000000001..
...11100000000000000..
0000000000000000000000
0000000000000000000000
```

e) resulting skeleton

```
0000000000000000000000
0000000000000000000000
0000000000000000000100
0001110000000000000000
0000000000000000000000
0000000000000000000000
```

It can be seen from the foregoing diagrams (a-e) that most of the information has been lost, the original line pattern has disappeared, and only a few pixels remain. In part this has arisen because parallel processing deletes opposite but neighbouring edge elements, as shown below:

```
000000000000000     000000000000000     000000000000000
011111111111110     0.............0     000000000000000
011111111111110     0.............0     000000000000000
000000000000000     000000000000000     000000000000000

       1                   2                   3
```

To overcome the problems indicated in the diagrams 1,2, and 3 above the SUB-PASS approach was introduced. Connectivity is retained by processing only the top edge of the object. Other edges are processed in later subpasses, and an iteration will consist of 4 (right, top, left, bottom) or 2 subpasses (right with top, left with bottom). The top subpass of a four subpass iteration is shown below:

```
000000000000   000000000000   000000000000   000000000000
011111111110   0tttttttttt0   0..........0   000000000000
011111111110   011111111110   011111111110   011111111110
000000000000   000000000000   000000000000   000000000000

     1              2              3              4
```

In fact in the above example all pixels remaining in 4 are skeletal and will not be removed in subsequent subpasses. The set of skeletal pixels remaining in 4 is the skeleton - but the reader will realise a directional bias of half-a-pixel has been introduced to the 'true' line which existed before scanning.

As we are concerned with the mapmaking applications of scanning and vectorizing we must consider whether systematically introducing the above mentioned half-a-pixel error to the position of a skeleton is important. For example frequently map scanning is performed with a scanner resolution of 0.1mm, so a half pixel shift is 0.05mm at map scale. In section B2.2.2 it states that, in digitizing contracts, the United States Government accepts a rmse of 0.13mm, or less, betwen the original map and the derived data. So the systematic shift of a half-pixel only becomes important when the scanner resolution is about 0.2mm or greater.

These classical algorithms are simple and easy to implement. If parallel processing hardware is available a fast approximation of the MAT is obtained, but few parallel processing implementations exist.

C4.1.1 LARGE WINDOW ALGORITHMS

Some classical algorithms may look at a window which is greater than 3x3. For example the Hilditch algorithm as implemented by Stefanelli and Rosenfeld [STEFANELLI, et al., 1971] is such an algorithm. It looks at the direct neighbours of two of the direct neighbour pixels, and also uses parallel processing. Another implementation of the Hilditch algorithm looks outside the matrix (window), but because it uses sequential processing it is too slow. This second implementation of the Hilditch algorithm does provide more acceptable skeletons than the classical algorithms, but its slow processing condemns it [NACCACHE and SHINGHAL, 1984].

C4.2 SKELETONIZING BY CONTOUR (EDGE) TRACING

Pavlidis [PAVLIDIS, 1982b] and Xia [XIA YUN, 1986] have developed contour tracing algorithms. In our experimentation we looked closely at these. (In the subsequent discussions an object will consist of ineterior and contour pixels.)

C4.2.1 THE XIA ALGORITHM FOR CONTOUR (EDGE) TRACING

This algorithm assumes that a pixel can be removed in the thinning process when its inner neighbour is in the interior of the object. The inner neighbour is defined as the closest neighbour in the direction of the bisector of the angle formed by the pixel and its two contour neighbours (c); or for each contour pixel of interest (i), a pixel called the inner neighbour (n) exists - as demonstrated in (a) to (c) below.

```
x n x      c . .      n c .      x n c
c i c      x i .      c i .      c i .
. . .      n . c      . . .      . . .
```

note: '.' = empty pixel and 'x' is a pixel whose value is unimportant

But, in the example below of a diagonal line, the Xia algorithm fails, and no thinning occurs.

```
        1 1 1                  1 1 1
      1 1 1                  1 1 1
    1 1 1      ----->      1 1 1
  1 1 1                  1 1 1
1 1 1                  1 1 1
                                         i.e. no change
```

C4.2.2 THE PAVLIDIS ALGORITHM FOR CONTOUR (EDGE) TRACING

The Pavlidis algorithm [PAVLIDIS, 1980] offered a new appproach to iterative thinning (peeling). The work of Pavlidis first consisted of a theoretical study of the problem of thinning curves. A mathematical point of view was adopted in this first phase - some geometrical properties of curves were looked at in the Euclidean plane and then on a discrete quantization grid. From this analysis, a new methodology for thinning was developed.

Pavlidis, like most of the other researchers, has designed criteria to distinguish between elements that can be thinned and elements that cannot and must not be thinned. But his criteria may be theoretically sounder than others. Pavlidis divided pixels into:

multiple pixels; and,
other pixels.

He proved that multiple pixels made up the final thin line (skeleton). His algorithm is based on differentiating between multiple pixels and other pixels for all those pixels belonging to the contour of an object. At each iteration, only multiple pixels are kept and other contour pixels are deleted. Each iteration creates a new contour, and the process is stopped when the new contour is the same as the previous one, that is when all contour pixels are multiple. A pixel is multiple if the following criteria are met:

1. it is traversed more than once by the object contour; or,
2. it has no neighbour in the interior of the object (all 8 neighbours in the 3x3 window are either contour elements or background elements); or,
3. it has a direct neighbour (P0, P2, P4 or P6) which is a contour element but which is not one of its neighbours in the contour.

The major advantage of this algorithm is that it is more robust than all classical thinning algorithms. (Also it is especially robust to criticism from other researchers!!)

The Pavlidis algorithm overcomes the following problems :

destruction of objects such as lines with uneven thickness;
excessive erosion of diagonal lines;
loss of small L and T patterns; and,
directional bias (each iteration consists of a unique pass and is more isotropic than the ones using Right, Top, Left and Bottom thinnings or any combination of those).

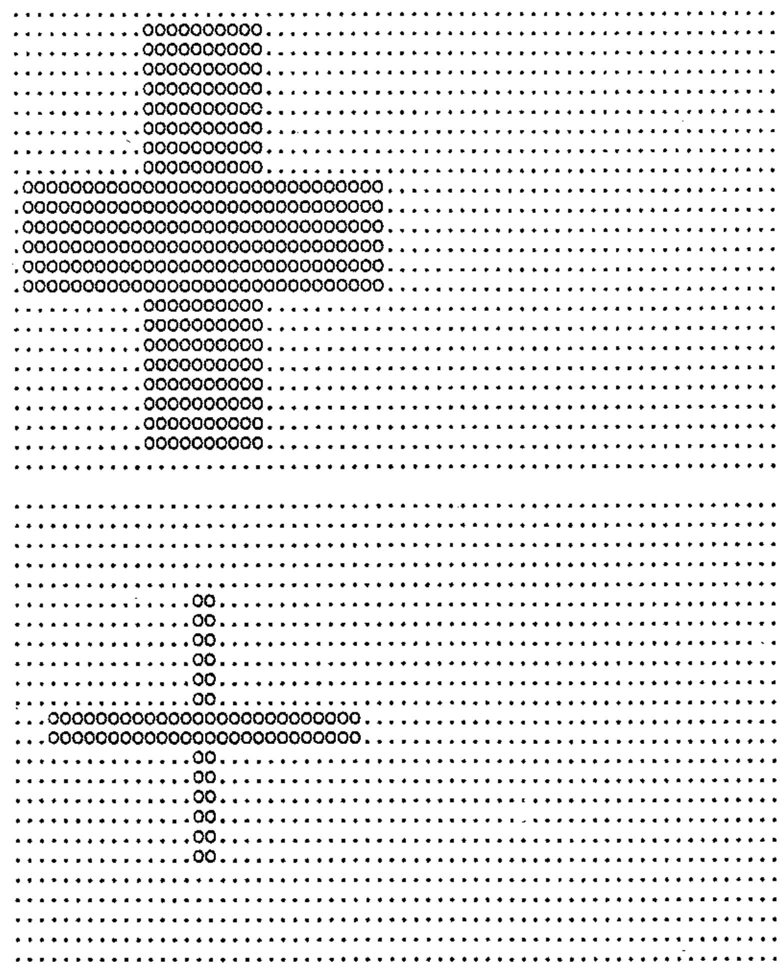
But, as usual, other problems remain, including:

node displacement : T junctions become Y junctions;
geometry alteration : L corners are distorted;
and, hole explosion : an isolated white pixel inside an area of black pixels causes a catastrophic result, since the hole is extended too much.;

The first two problems especially occur when a thick line intersects a thin line. But these rather bad results need not cause too much despair as classical algorithms cannot overcome the problems either.

Now a new problem appears! This algorithm cannot produce lines with unit thickness everywhere (see FIGURE 18), due to the use of the second and third criteria listed above. Pavlidis designed a new procedure which he called POSTPROCESSING in order to reach the necessary unit thickness; this task is actually a single iteration thinning which is much simpler than iterative thinning but unfortunately seems unsuccessful. We instead developed a modification of Pavlidis' approach which converts an incompletely skeletonized image into a perfect skeleton. It does this in one pass in which the image is scanned once from upper left to lower right. (Other directions are, with some minor modifications, also possible.) When an object pixel is detected the neighbours are inspected and, essentially, if the pixel has a crossing number (CN where CN - or the crossing number - is the number of background to object conversions in the neighbourhood) of 1 and more then one object neighbour pixel, it is discounted as a skeleton pixel. After this a new value is given to the pixel in accordance with its crossing number (2 for CN<3 and 3 for CN>2). Then, one at a time, the North and the West neighbours are deleted if they have a CN smaller then 3. The CN is computed again and it is compared with the initial CN. If these are different the deleted pixel is reset as an object pixel.

FIGURE 18. THE RESULT OF PAVLIDIS THINNING, SHOWS THAT A TWO PIXEL WIDE LINE CANNOT BE FURTHER THINNED.



C4.2.2.1 DEFICIENCIES OF THE PAVLIDIS ALGORITHM

We have tested the Pavlidis algorithm, to examine its options and defincies. Five of these are outlined below.

1) Corner Option

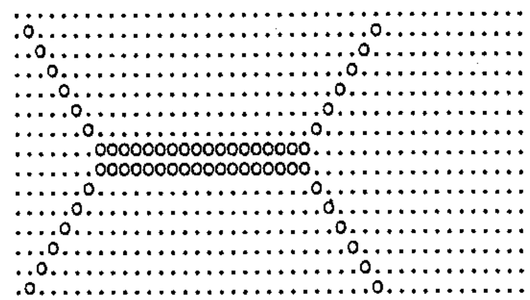
According to Pavlidis [PAVLIDIS, 1980] a test can be used to control the creation of skeleton branches. This is described as the 'corner option'. A corner pattern produces a skeleton branch, as shown:

```
0 0 0 0 . .      0 0 0 0 . .      0 0 0 0 . .
0 1 1 1 . .      0 1 0 0 . .      0 1 0 0 . .
0 1 1 1 . .      0 0 1 1 . .      0 0 1 0 . .
0 1 1 1 . .      0 0 1 1 . .      0 0 0 1 . .
. .              . .              . .
```

or as in FIGURE 19.

FIGURE 19. THIS SHOWS THE RESULT OF PAVLIDIS' THINNING WITH THE CORNER OPTION (a) AND WITHOUT THE CORNER OPTION (b)

(a)



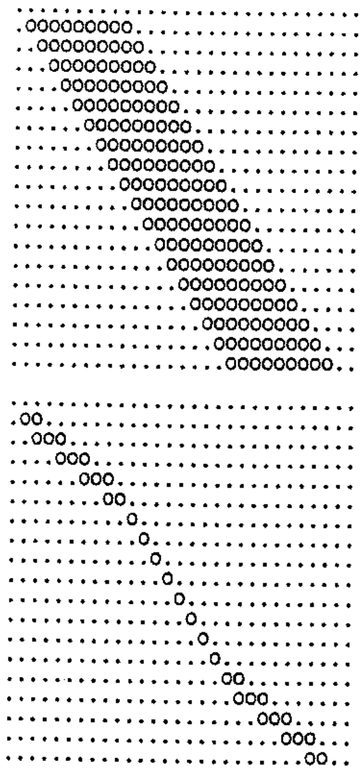
(b)

Such an effect is described by Stefanelli and Rosenfeld [STEFANELLI, et al., 1971] as originating from the absence of noise cleaning conditions! The 'corner option' produces an algorithm which is very noise sensitive (in terms of contour, or edge, noise) and which gives spurious tails whenever there is a corner. We have found this option deficient.

2) Small convexities

A similar case of spurious tails occurs with the pattern shown in FIGURE 20, where small convexities evolve.

FIGURE 20. THE PROBLEM OF SMALL CONVEXITIES BEING INTRODUCED BY THE PAVLIDIS ALGORITHM.



Examples of this problem can also be seen in FIGURES 23, 24, 25.

3) At the image edge

As with every approach, what do we at the outer frame of the image, where there is inadequate information?

4) Behaviour of holes

The 'explosion' of holes can also be seen in FIGURES 21.A and C (before the explosion) and FIGURES 21.B and D (after the explosion).

FIGURE 21 THE CATASTROPHIC EFFECT OF NOISY HOLES

FIGURE 21.A BEFORE AN EXPLOSION

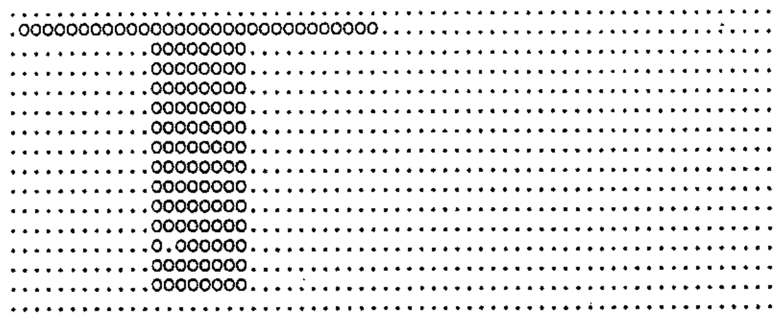


FIGURE 21.B AFTER AN EXPLOSION - THE EFFECT ON 21.A

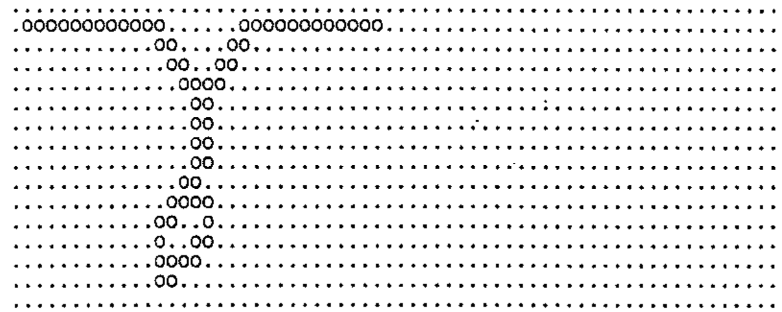


FIGURE 21.C BEFORE AN EXPLOSION

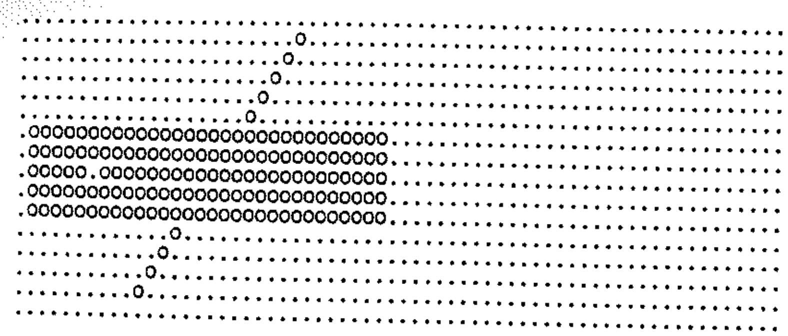
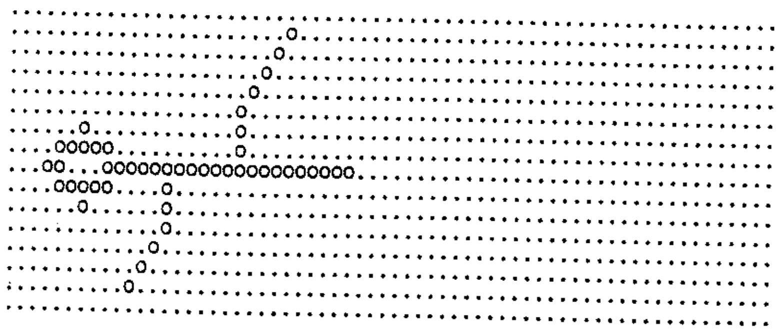


FIGURE 21.D AFTER AN EXPLOSION - THE EFFECT ON 21.C



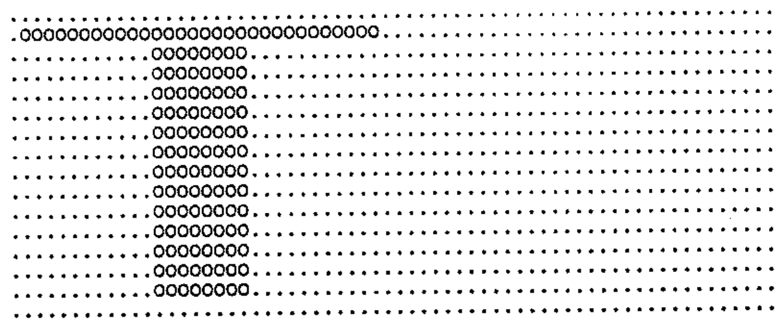
5) Excessive erosion causing node displacement and corner distortion.

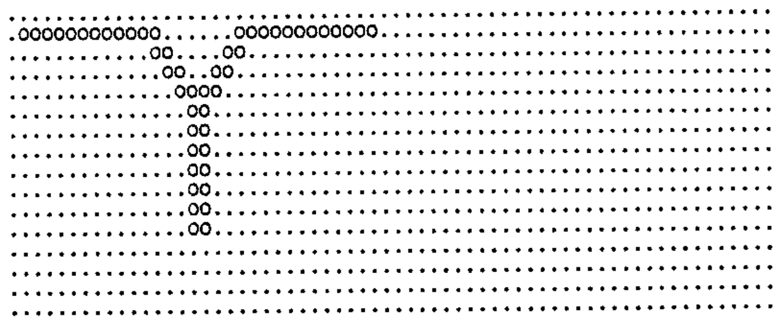
This is demonstrated in FIGURE 22.

FIGURE 22. THE PROBLEM OF EROSION AT JUNCTIONS

FIGURE 22A. A T JUNCTION BEFORE EROSION

FIGURE 22B. A T JUNCTION AFTER EROSION





It should be noted that examples of all five problems can be seen in the FIGURES 23, 24, and 25. These three figures represent data sets before and after processing by the Pavlidis algorithm.

FIGURES 23, 24, and 25 SHOWING THE SEVERAL EFFECTS OF THE PAVLIDIS ALGORITHM

FIGURE 23A BEFORE PROCESSING THROUGH THE PAVLIDIS ALGORITHM

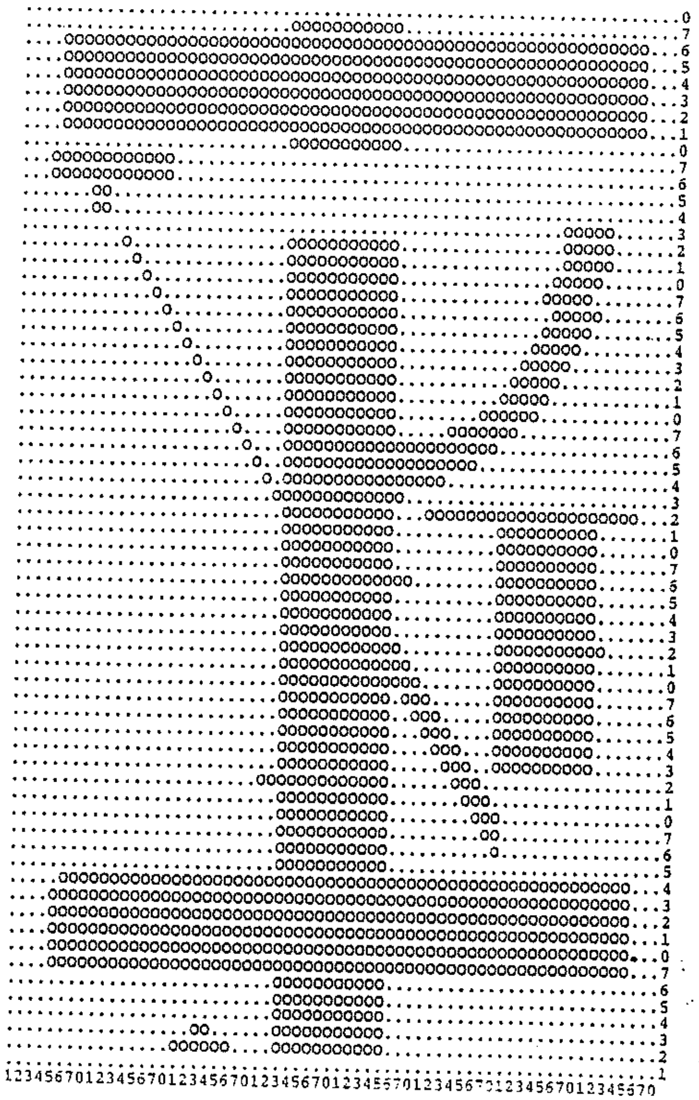


FIGURE 23B AFTER PROCESSING THROUGH THE PAVLIDIS ALGORITHM (showing T-displacement, Y-displacement, Line-end shrinking)

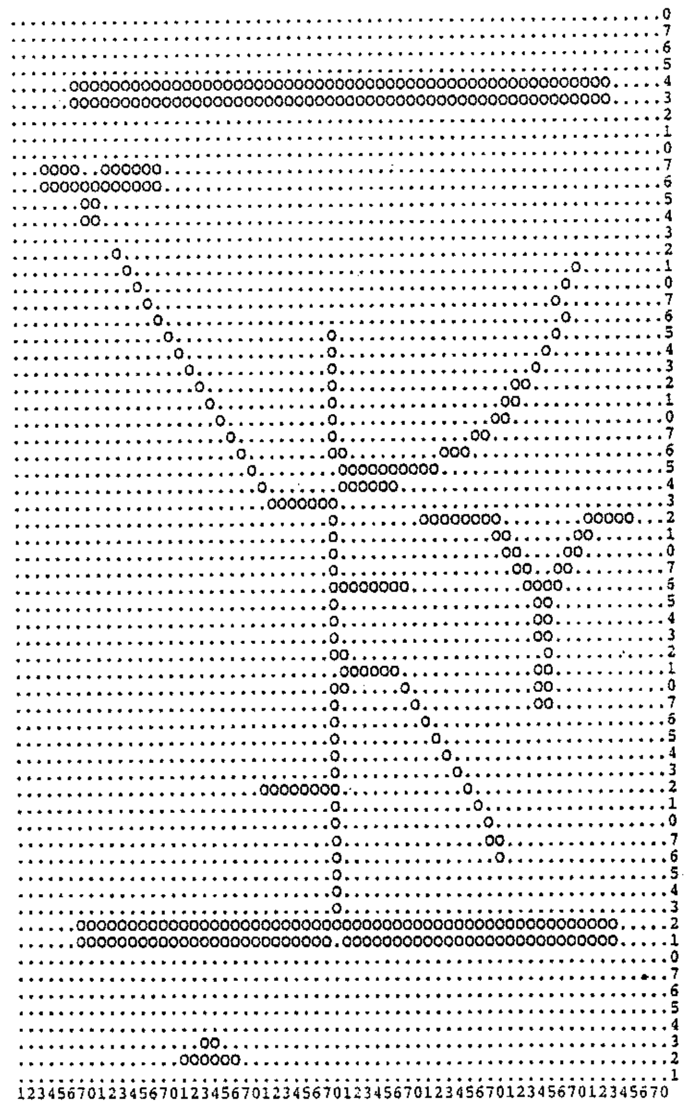


FIGURE 24A BEFORE PROCESSING THROUGH THE PAVLIDIS ALGORITHM

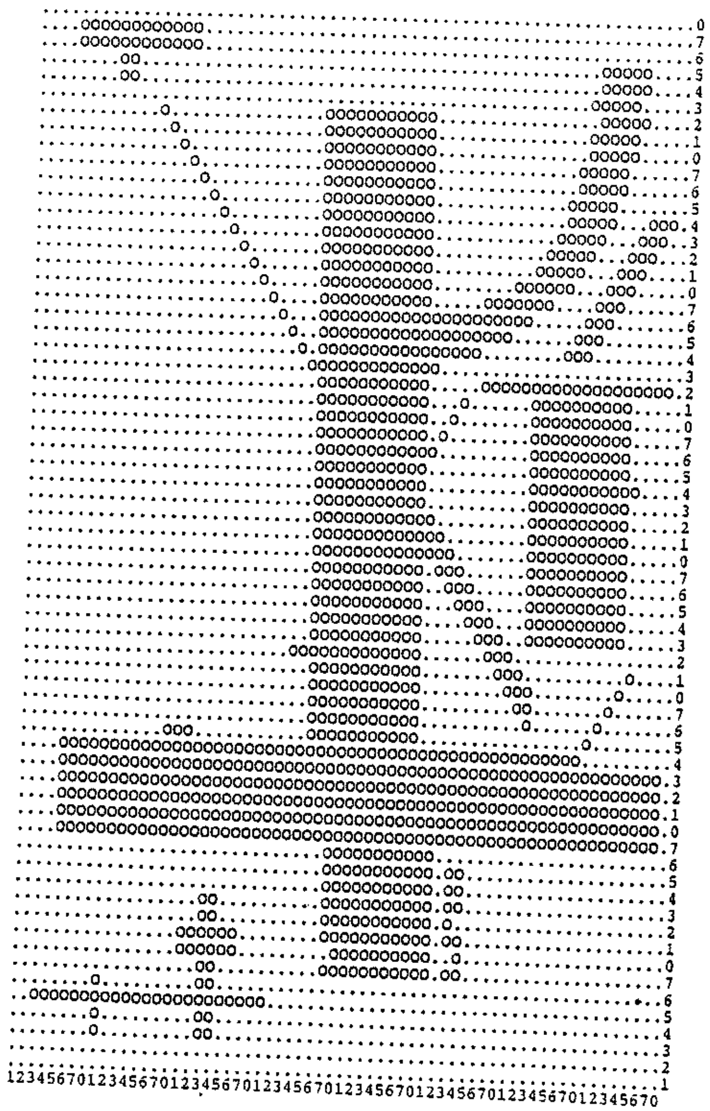


FIGURE 24B AFTER PROCESSING THROUGH THE PAVLIDIS ALGORITHM (showing T-displacement, Y-displacement, Line-end shrinking, 2-pixel wide lines, Noisy tails)

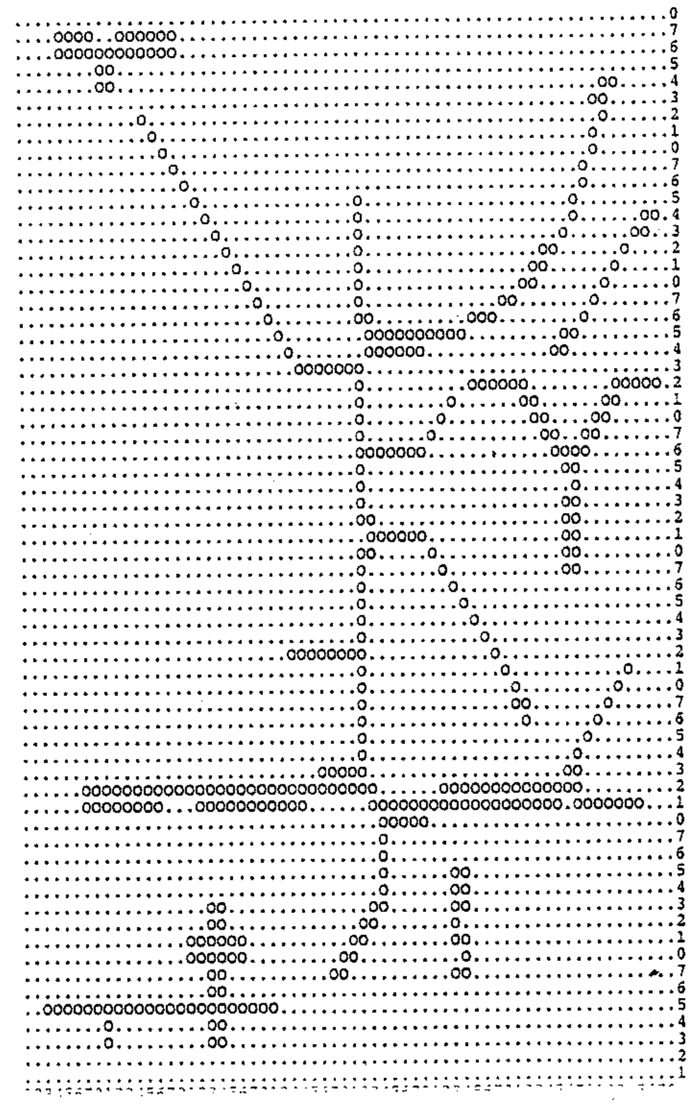


FIGURE 25A BEFORE PROCESSING THROUGH THE PAVLIDIS ALGORITHM

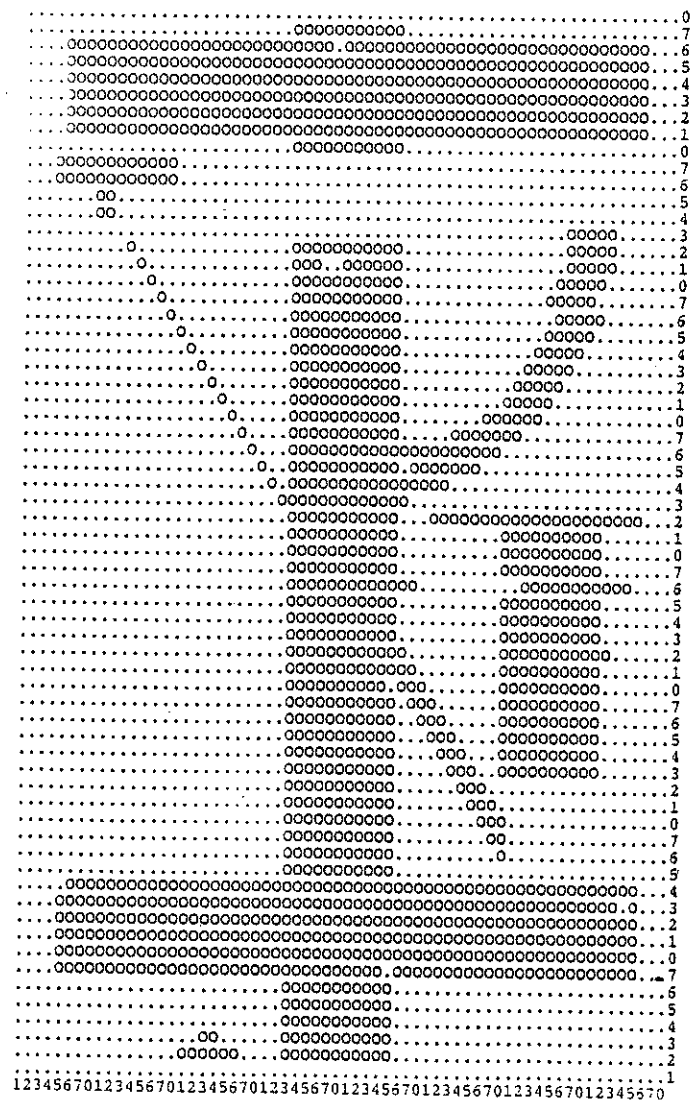
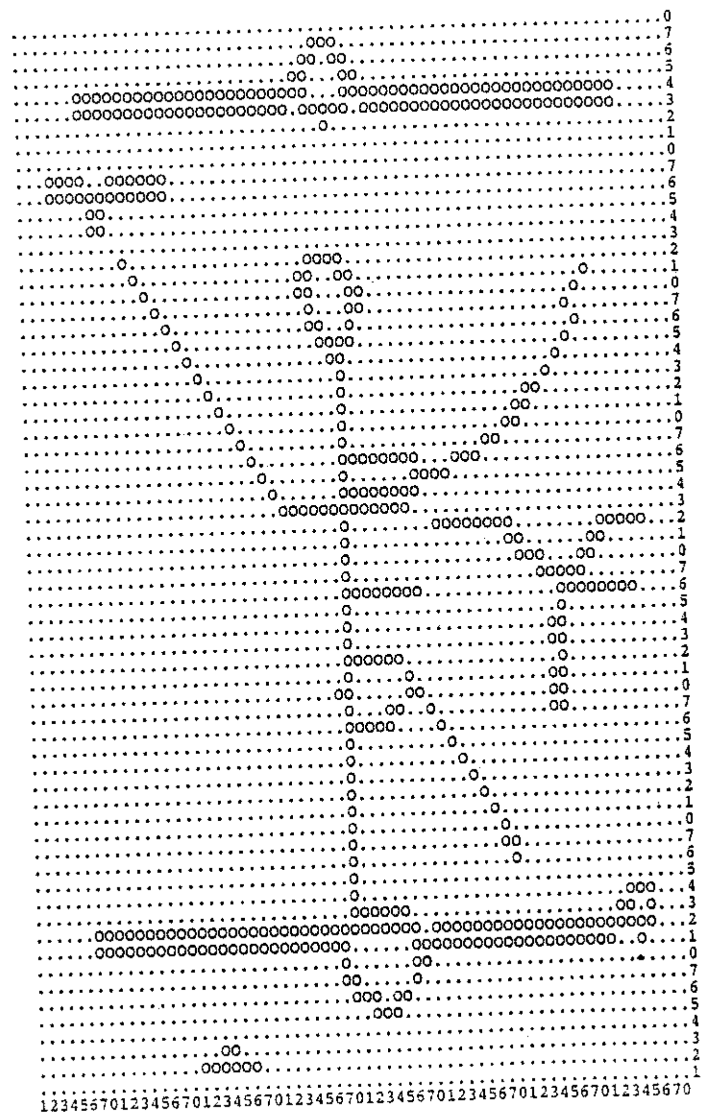


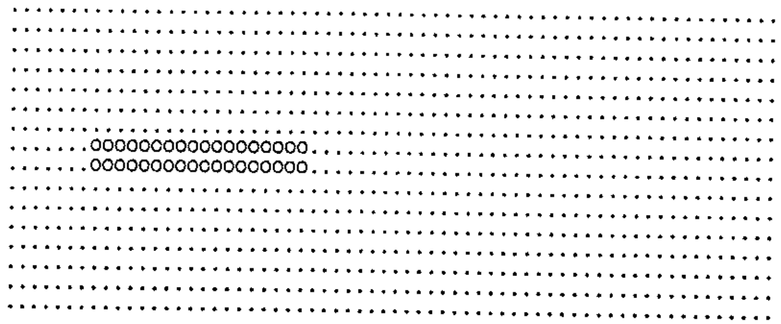
FIGURE 25B AFTER PROCESSING THROUGH THE PAVLIDIS ALGORITHM
(showing small holes, contour noise)



C4.2.2.2 IMPROVEMENTS OF THE PAVLIDIS ALGORITHM

1) Corners

Corner information can be obtained by considering the contours in the original image (which is Pavlidis' default), and not to use the corner option.



2) Small convexities

We propose looking at one pixel called the extended inner neighbour, this pixel is X in the diagrams below:

```
0 0 1 X
0 1 1
0 0 0
```

X is the extended inner neighbour of the central pixel, because it falls on the bisector of the angle formed by the central pixel (the pixel of interest) and its two contour neighbours.

We propose the following action:

- if X is in the interior of the object, then all 3 pixels - the central one and its 2 contour neighbours - will be deleted. If not, they are kept.

```
    1        0
00111    00010
0111     0000
000      000
```

This test has no theoretical foundation, but, subjectively, it gives better looking skeletons.

3) At the image edge

The outer frame of the image poses problems as soon as there are object pixels in it. This is because we do not know what is outside the image, but the possibilities are:

1. nothing or background;
2. data which is a mirror image of that inside;
3. data which is obtained by extrapolation; and,
4. data from another image - this is especially true when an image is processed by patches because the computer memory is not sufficient to store the whole image.

Filling the outside of the image with '.' - pixels is the easiest way to remove this annoyance. But in patch processing, it will be more difficult to merge data sets coming from adjacent patches because of excessive thinning.

```
................|................
1111111111111111|1111111111111111   original 2 patches
1111111111111111|1111111111111111
1111111111111111|1111111111111111
................|................

................|................
................|................
111111111111111.|.111111111111111  2 patches showing excessive
................|................  thinning at the ends
................|................
```

(in these diagrams 1 = object, and . = the absence of object)

Patch processing cannot use connectivity information between adjacent patches, unless all patches are simultaneously processed by an equal number of processors exchanging connectivity information when necessary. This is almost equivalent to the theoretically possible (but practically impossible) situation of 1 single processor working on the whole image. An ad hoc and not very elegant attempt has been tried, using either 'reflected' data or 'extrapolated' data. But the user of thinning algorithms must be vigilant when having to deal with patches and data that overlap several of these patches.

4) Behaviour of holes

White noise removal deletes noisy unwanted holes. These techniques are further discussed later.

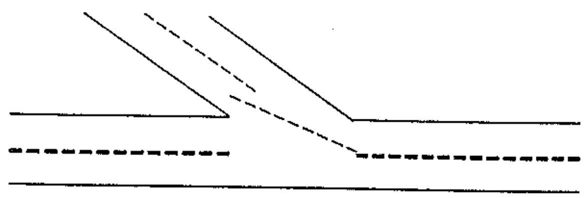
5) Excessive erosion causing node displacement and corner distortion.

This problem of excessive erosion is addressed in the next section, and it will be shown it is solved by the TON algorithm.

C5. SKELETONIZING WITH NODE IMPROVEMENT

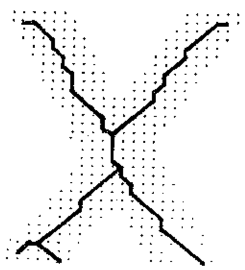
In general skeletonization should result in a picture which represents the object by its medial line [STEFANELLI et al., 1971]. But representing an object by its medial line has disadvantages when used in the vectorization process of a map. When vectorizing a simple linear feature like a road, the medial line is what we want. But as soon as the road reaches a cross road problems arise. In FIGURE 26, a road with a cross road, together with the medial line of this object is shown.

FIGURE 26. ROAD WITH A CROSS ROAD



It can be seen that where the roads intersect, the node region, the concept of the medial line is ambiguous, and does not represent the characteristics of the object. What does represent the characteristics of the object are the medial lines of the separate objects (the two roads). However, in some cases even this is not sufficient. FIGURE 27 gives an example of such a case. What is shown in this figure are 4 line features which intersect in one junction. However, the medial lines of these four line features do not intersect in one point. This is the so-called 'hourglass effect'. We can see that it is impossible to represent the characteristic features of the object by any medial line.

FIGURE 27. THE HOURGLASS EFFECT



Thinning intersections neither results in the medial line, nor in the correct representation of the object. This phenomenon we call node displacement.

It should be stressed that node displacement is a direct result of the non-existence of the medial line of an object at intersections. This means that node displacement does not occur because a bad thinning algorithm is used, but instead that thinning is a bad concept for obtaining a skeleton which represents all characteristic features of the original raster image. From this it can be concluded that another concept should be applied to obtain the desired result. However, an extensive literature search did not reveal any alternatives. Therefore, in anticipation of a good concept, we decided to look for ways to change the behaviour of thinning algorithms near nodes. The result is an algorithm we call TON (Thinning Of Nodes).

In non-cartographic literature little interest is shown in the problem. Deutsch mentions the subject because it makes it impossible to distinguish between a B and an 8 in Optical Character Recognition [DEUTSCH, 1975]. Tamura only recognizes the problem for intersecting lines of equal thickness; a situation in which the displacement is only one pixel [TAMURA, 1976].

In the cartographic literature the problem is recognized [PEUQUET, 1981; DRUMMOND, 1986; PICKERING, 1988] but no attempt is made to solve the problem. In the course of our project we were confronted with two studies [DOMOGALLA, 1984; ESPELID, 1988] dealing with the problem. Because the methods of Domogalla and Espelid are quite similar, we restrict ourselves to a brief description of the method developed by Domogalla.

These next sections (C5.1 to C5.6) are not for the faint-hearted! We begin by revising (C5.1) Domogalla's method for avoiding the problem of node displacement, which is the main component of the 'distortion of detail' listed in C1. Then an analysis of the origins of the node displacement is presented (C5.2) and it is shown that this is caused by the formation of spurious triangles at junctions during thinning, and 'over-thinning' at the non junction (or 'other side') of a junction. Methods for preventing both the formation of the spurious triangle and the 'over thinning' which have been developed by us are presented.

In order to work, these (and also Domogalla's) methods require the successful detection of junction/node areas, and an approach, based on vector analysis (C5.5) is presented. Finally in section C5.6 the TON (Thinning-of-Nodes) algorithm is presented. This incorporates ideas developed in the sections C5.2 to C5.5 to produce a thinning algorithm which does NOT introduce node displacement.

As already stated, node displacement is a 'cartographic' problem - or at least a problem of those who are concerned with extracting the highest metric accuracy possible from existing map documents. We must look look to ourselves, within the mapping professions, to find solutions to this problem

C5.1 DOMOGALLA'S METHOD OF PREVENTING NODE DISPLACEMENT

Domogolla's method starts with calculating the Medial Axis Transform (MAT) of an image (see [DOMOGALLA, 1984] for a description of this calculation). From this MAT the skeleton of the image is found. In the skeleton, pixels are divided up into four groups:

1) line points;
2) line end points;
3) node points; and,
4) isolated points.

Domogalla then processes line segments which are defined as groups of connected line point pixels between two groups of node point pixels. Line segments can be described by the function D = f(B), where D is the MAT value of that pixel and B the distance along the segment from one of the two nodes. Maxima in this function occur at node points and minima well within the line segment and away from the nodes. On the basis of these maxima and minima the 'division point' (as described in the next paragraph) in the line segment is determined.

Division points lie in the line segment between the minimum and maxima, and where maximum MAT value change occurs. For each division point border points are computed. Border points are those pixels, which are on the border (contour, edge) of the original object and have minimum distance to the division points. In this way for each division point two border points are computed. Then this pair of border points together with pairs of border points from neighbouring line segments are connected by a polygon in a clockwise direction.

These polygons delimit the node regions. During the vectorization process these node regions are not processed.

When the whole image is vectorized, vectors running into node regions are extended to intersect at the node [DOMOGOLLA, 1984, pp.70-101].

C5.2 THE ORIGINS OF NODE DISPLACEMENT

FIGURE 28 gives a raster representation of FIGURE 26. The situation which exists after two thinning iterations is depicted in FIGURE 29. It is assumed that FIGURE 28 is only a part of the image which is thinned, so that no pixels are line ends.

FIGURE 28. A RASTER REPRESENTATION OF FIGURE 26 (pixels represented as +,x are referred to in later discussion)

```
          *****
           *****
            *****
             +*****
**************xx**********
**************************
**************************
**************************
**************************
**************************
**************************
```


FIGURE 29. A REPRESENTATION OF FIGURE 28 AFTER 2 THINNING ITERATIONS (pixels represented as A,B,C,D are referred to in later discussion)

```
               *
                *
                 *
                  *
                   A
                  *D*
                 *****
**************************
*******************B***C**
**************************
```

Subsequent iterations will skeletonize the remaining unskeletonized part of the diagonal branch, which (as can be seen from FIGURE 29) runs perpendicularly into the horizontal branch. Because the remaining unskeletonized part meets the horizontal line as a perpendicular the resulting node (at B) will be displaced to the left of the position at which one would hope to find it (at C). From now on we will call this kind of displacement y-displacement (or DY), because it distorts the y shape of the configuration.

Considering the size of this displacement (DY), from the triangle formed by the pixels ABC in FIGURE 29, we see:

$$DY = (d1 + resolution)/(\tan i)$$

Where:

i	= the angle of incidence of the road (angle ACB);
d1	= the half the line width of the horizontal branch (length DB) (before skeetonizing);
resolution	= the resolution with which the image was scanned, that is the side of one pixel (length DA).

After further iterations, however, something else is going to happen, as shown in FIGURE 30. Not only the diagonal branch is distorted, but also the horizontal one. The determined position of the node is 'pulled towards' the diagonal branch. We will call this displacement T-displacement (or DT), because it distorts the T-shape of the branches.

It can be shown that the T-displacement satisfies the following equation:

$$DT = (d1 - d2) / 2;$$

where:

d1 = half the line width of the horizontal branch;
d2 = half the line width of the diagonal branch.

FIGURE 30. A REPRESENTATION OF FIGURE 29 AFTER FURTHER THINNING ITERATIONS

```
         *                                 *
          *                                 *
           *                                 *
            *         goes to                *
             *                                *
            ***                              ***
           *****                            *****
     ****************                *********  *******
```

Because Y-displacement occurs always in a direction perpendicular to the T-displacement we can conclude that the total displacement equals:

$$D = SQRT(DY^2 + DT^2)$$

We have supposed straight branches. Considering curved branches, mathematically we can only assume:

i) that the curvature just outside the node region is continued inside the node region, or
ii) that the curvature inside the node region becomes zero and consequently that the angle of this part of the medial line equals that just outside the node region.

When we make the first assumption we can say that the extra node displacement due to the curvature of the branches is proportional to the difference in curvature of the two branches and inversely proportional to the angle of incidence. However, it is theoretically possible that the medial line of two intersecting branches will never meet, which does not accord with reality. The first assumption is unacceptable. This leaves us the second assumption, and with this the above equation of node displacement still holds. Thus node displacement of branches with a curvature different from 0 is the same as those of uncurved branches.

From the preceding figures it can be seen that the y-displacement arises because the last part of the diagonal branch is first thinned to a triangle and then this triangle is thinned to a skeleton perpendicular to the horizontal branch. T-displacement originates from the fact that the diagonal branch, after several iterations, is also thinned from the 'other side' of the horizontal branch. This is partly the result of the existence of the triangle.

However even if we suppose the above mentioned triangle does not exist, T-displacement can occur. Considering a junction with a horizontal branch one pixel thick, but an incoming diagonal junction which is thicker (as shown in the following diagram):


```
    AxxxB
     AxxxB
      AxxxB
xxxxxCCCxxxxx
```

At the first thinning pass the horizontal branch is declared skeletal and the diagonal branch is thinned. The result of this is that thinning at the A, B, and C pixels occurs giving us T-displacement. (The C pixels may be considered to be pixels at the 'other side'.)

(In the foregoing, when describing a certain configuration of pixels, we have depicted this configuration in a figure and have been talking about horizontal and diagonal branches. It must be clear however, that the configuration described is rotation invariant; the terms 'horizontal' and 'diagonal' are only used to make the distinction between the ongoing and the terminating branch. We will go on doing this, so the reader is asked in these cases to bear in mind that the given example also stands for rotated versions of it.)

Of course other configurations of branches exist, but as almost all existing thinning algorithms only take into account the 8-neighbourhood pixels, only those configurations which imply different 8-neighbourhood configurations will yield different results. The only configuration which differs fundamentally from the one dealt with so far is the case where both the branches go on after the node. In this case no T-displacement will occur because thinning from 'the other side' will not occur. Y-displacement will take place in the described way. In this case we actually are dealing with two nodes on the same place which are being displaced in a different direction. This results in the previously mentioned 'hour glass' effect at X-crossings. It has the same origin as the normal Y-displacement.

We will now describe our solutions for the prevention of node displacement; first Y-displacement and then T- displacement.

C5.3 PREVENTING Y-DISPLACEMENT OF A NODE

From the foregoing we can conclude that Y-displacement can be prevented if creation of the triangle can be prevented. This triangle forms because the two pixels marked x in FIGURE 28 are not deleted. If we modify the thinning algorithm so that these are deleted, Y-displacement is prevented. However in so doing the incoming branch must be extended by a short branch having an incoming angle - described here as the extension angle - of 45 degrees. This extension angle is only acceptable if the incoming branch truly approaches the horizontal branch at this angle - otherwise node displacement has arisen. Ideally the extension angle should equal the intersection angle. But in 8-metric space not all angles are possible. There follows a list showing the relationship between neighbourhood size and the possible angles:

neighbourhod size	possible angles (in degrees)
3 x 3	0, 45, 90
5 x 5	0, 26.6, 45, 63.4, 90
7 x 7	0, 14.0, 26.6, 36.9, 45, 53.1, 63.4, 76.0

Clearly, to avoid complexity, we must accept a limited choice for the extension angle. The intersection angle is divided up into certain 'angular regions', and with each region a certain extension angle is associated. As a result some node displacement will still arise.

This displacement must be quantified - and to do so we return to the equation:

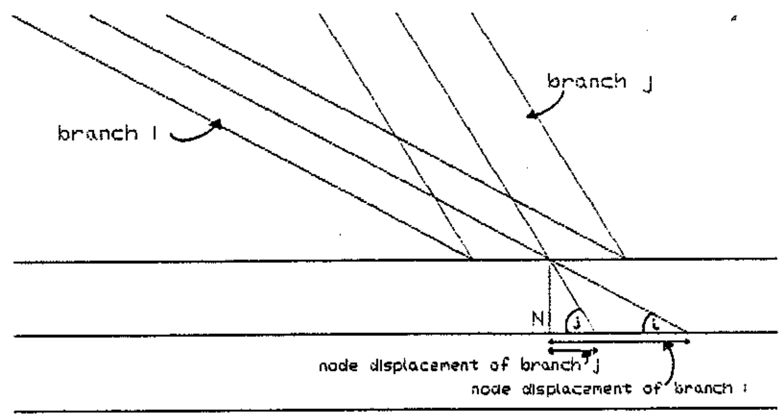
$$DY = (d1 + resolution)/(\tan i)$$

first given in section C5.2.

This equation gives the relationship between the intersection angle and the resulting y-displacement. If a branch has an intersection angle of i and is extended with an angle j we can derive the resulting node displacement as follows:

Consider two branches, one with an intersection angle i and one with an intersection angle j. If we thin according to a normal thinning algorithm a node will be yielded, which will in both cases be in the same position (N). As can be seen in the diagram below, the node displacement for the branch i is greater than

FIGURE 31. TWO INCOMING BRANCHES (i and j) MEETING A HORIZONTAL LINE



that for the branch j, and we can also say that the node displacement resulting from using an extension angle of i is:

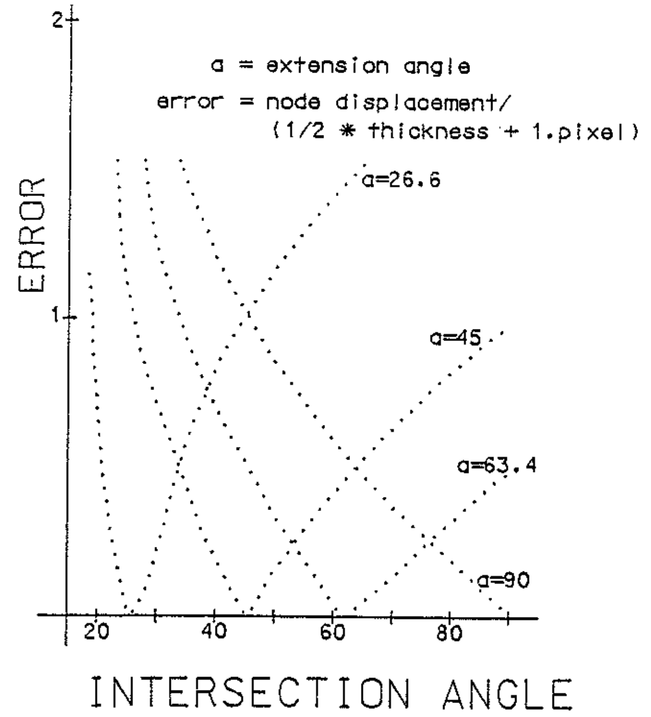
$$DY = (d1 + resolution)(1/\tan(j) - 1/\tan(i)) = A/\tan(j) + B$$

in which A and B are constants.

Figure 32 depicts the preceding relationship graphically for the extension angles (a) possible in a 3 x 3 and a 5 x 5 neighbourhood. In this figure the graph of an extension angle of 90 degrees depicts the situation in normal thinning algorithms, and in this case all the extension angles are extended by angles of less than 90 degrees. When we introduce more extension angles and we call the related graphs f1 .... fn we can say that the resulting node displacement equals: MIN(f1 ... fn). Thus it is possible to quantify the consequences of certain choice of extension angle.

FIGURE 32 DIAGRAM SHOWING THE ERROR INTRODUCED BY CERTAIN EXTENSION ANGLES

Where the user selects the extension angles to be used, and reads off the graph the node displacement error (for the TRUE intersection angle) created by selection of the particular extension angle.



From the figure we can see that the node displacement is most serious at very small intersection angles. This would make a very small extension angle necessary. Since this needs operations on a very large neighbourhood this is not an acceptable solution. However an analysis of various nations' topographic maps has shown that intersection angles of less than 30 degrees do not occur, so only intersection angles larger than this need to be considered. The table below lists the maximum error resulting from the choice of the two extension angles 63.4 degrees and 45.0 degrees depicted in the above figure.

TABLE 5. INTERSECTION ANGLE REGIONS AND THEIR EXTENSION ANGLES

Nr. of regions	regions	extension angles	Maximum error
2	30 - 63.4 degrees	45 degrees	0.70 pixels
	63.4 - 90 degrees	90 degrees	0.50 pixels
4	30 - 36.9 degrees	26.6 degrees	0.50 pixels
	36.9 - 53.1 degrees	45 degrees	0.25 pixels
	53.1 - 76.0 degrees	63.4 degrees	0.25 pixels
	84 - 90 degrees	90 degrees	0.25 pixels

From the above table it can be seen that the last case yields a maximum errror which is approximately 3/4 of the first case. However, the last case is more complex and more computationally demanding. In the first case the maximum error appears at an angle of 30 degrees. This situation is depicted in the FIGURE 33 below. From this we can see that it yields a displacement perpendicular to the diagonal branch of:

$$d_p = DY (\cos(90 - i))$$

and since i = 30 degrees then $d_p = 0.5 * DY$

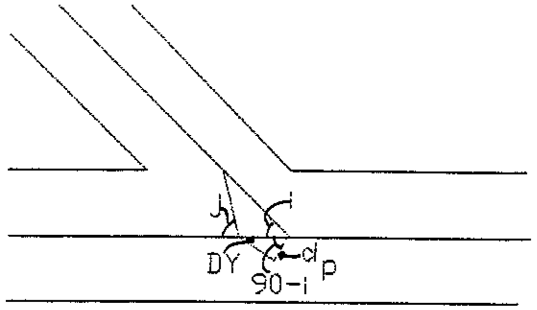
but since DY = (d1 + resolution) * 0.7

$$= d_p$$
$$= 0.5 * 0.73 * d1 + 0.5 * 0.7 * resolution$$
$$= 0.35 * d1 + 0.35 * resolution$$

An equal argument shows that for the second case:

$$d_p = 0.28 * d1 + 0.28 * resolution$$

FIGURE 33. DIAGRAM SHOWING Y-DISPLACEMENT



Cartographically a node displacement is acceptable if the node is positioned inside the region occupied by both the intersecting branches [BECK, et al., 1986]. This requirement is not met if the horizontal branch is about three or more times as thick as the diagonal branch. When the horizontal line is thicker than this 4 regions instead of two will have to be applied, to select the extension angles.

C5.4 PREVENTING T-DISPLACEMENT

T-displacement arises because thinning occurs on three sides of a junction. It can be prevented because, as soon as the horizontal branch is skeletal outside the node region, the pixels at 'the other side' of the skeleton are declared skeletal. As this is done only thinning from the proper side is possible and T-displacement will not occur.

C5.5 FINDING JUNCTIONS

The approach described above (section C5.4) requires a good method of detecting intersections. In a skeletal phase this is quite easy, but in a non-skeletal phase it is more complicated. In the latter phase we will call intersections nodes, and to distinguish intersections in a skeletal phase from nodes, we use the word junction. The pixel marked + in figure 28 we call the junction pixel, and finding junctions is a matter of finding junction pixels.

The most obvious way to find junctions is to compare the neighbourhood of every pixel in the image with a predefined configuration, i.e. a method of template matching. When we want to make a 100% distinction between junctions with intersection angles of less than 63.4 degrees and those with more (as is required in section C5.3) we need a 25-neighbourhood, but such a large neighbourhood yields unacceptable computation times. So we sought another method.

First we considered the configuration of Freeman codes around the junction pixel. FIGURE 34 depicts all the possible configurations of pixels together with their sequence of Freeman codes when following the contour in an anti-clockwise direction.

One possibility is to follow the contour of an object looking for a sequence of 5 Freeman codes resembling a sequence in FIGURE 34. We implemented this using only two templates - the templates 66544 and 77544 of FIGURE 34.

FIGURE 34. POSSIBLE JUNCTIONS

a: >= 60 degrees
b: < 60

```
OOOXX  OOOXX  OOXXX  OOXXX  OOOXX  OOXXX  OOXXX  OOXXX
OOOXX  OOOXX  OOOXX  OOOXX  OOOXX  OOXXX  OOOXX  OOXXX
OOOXX  XOOXX  OOOXX  XOOXX  XXOXX  OOOXX  XXOXX  XOOXX
XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  OXXXX

66544  66543  76544  76543  66534  67544  76534  67543
```

a. THE PIXEL DEPICTION OF INTERSECTIONS GREATER THAN 60 DEGREES, WITH THEIR FREEMAN CODES

```
OXXXX  OXXXX  XXXXX
OOXXX  OOXXX  XXXXX
OOOXX  XOOXX  OOXXX
XXXXX  XXXXX  XXXXX

77544  77543  60754
```

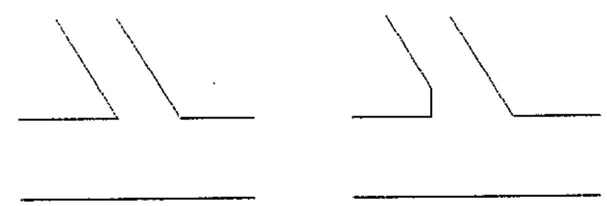
b. THE PIXEL DEPICTION OF INTERSECTIONS LESS THAN 60 DEGREES, WITH THEIR FREEMAN CODES

The results were good on noise free images; in reality there are no noise free images. One can try to reduce the amount of noise, but, as yet, no algorithm is known giving a 100% guarantee of deleting all noise in an image. Noise near junctions is especially hard to detect and junctions are places where noise is very likely to occur, as this is where the drafting tool, while drawing the image, stopped for an instant, undertook a sharp change in direction or was raised or lowered from the paper. Furthermore, the same configuration of pixels is noise at one junction, while representing the actual form of the junction at another (See FIGURE 35).

This means we can not make any predictions about the configuration of pixels near a junction. We only can say that there must exist two branches which meet at a certain angle. That is why we must look for contour parts which form a certain angle. We can make this operational by looking for sequences of Freeman codes equal to the first two and the last two codes in the sequences shown in FIGURE 34. (Of course the distance between the members of one pair of sequence parts must be under a certain predefined limit.) If these are found then the pixels which prevent the junction from having the ideal shape, whether they are actual noise or not, are removed.

FIGURE 35. UNCERTAINTY ABOUT NOISE

Original image:



Scanned image:

```
       xxxx                         xxxx
        xxxx                         xxxx
        nxxxxx                        xxxxx
xxxxxxxxxxxxxxxxxxx           xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx           xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx           xxxxxxxxxxxxxxxxxxxx
```

(n is a noise pixel)

However this method has one severe drawback. It only works when the noise contour does not resemble one of the predefined sequence parts. As the sequence parts consist of only two Freeman codes, it is quite likely that the resemblance arises - so longer sequences are needed and then the number of templates needed increases.

A possible solution is to use vectors instead of Freeman codes. We can do this by vectorizing the contour. Then an algorithm just looks for vector end points. If at a point (point 1) the preceding vector (vector 1) has a length above a certain limit and there exists a vector further on in the chain (vector 2), which satisfies the following criteria:

1) it has a length which is above a certain limit;
2) the distance from its startpoint to point 1 does not exceed a certain limit; and,
3) there exists no vector with a length which exceeds the limit mentioned under 1) between point 1 and the startpoint of the vector;

then the intermediate vectors are examined. If there are intermediate vectors the junction is ideal already and we just have to store the coordinates of point 1. If there exist intermediate vectors we distinguish two cases:

1) the directions of these vectors fall inside the limit formed by the directions of vector 1 an vector 2; and,
2) they do not.

In the first case all contour pixels between vector 1 and vector 2 are deleted. New contour pixels are generated and the contour from the first pixel of vector 1 to the last pixel of vector 2 is re-vectorized. Of these vectors the first is called vector 1 and the last vector 2. When there are still vectors between vector 1 and 2, other than one vector of unit length, the process is repeated - starting from deletion of contour pixels. Of course we run the risk of excessive erosion. Although convinced that this chance is zero, a test can be implemented which prevents excessive erosion.

The test can run by adding a boolean field 'visited' to the record of contour pixels. When a pixel is put in the contour chain it gets a different value in the image, to discriminate it from the non-contour pixels. When a pixel, which already has the changed value, is put in the contour, the 'visited' value is turned to TRUE. Deletion of a pixel with 'visited' value of TRUE will yield excessive erosion and we will have to stop the process. Then of course the skeleton is distorted. To restore the original form we must have kept track of the pixels so far deleted. If we have done this, restoring the image is just a matter of returning stored pixels to their original value.

In the second case just the opposite is done; all background pixels neighbouring the contour between vector 1 and vector 2 are set to 'object', the pixels between the start point of vector 1 and the end point of vector 2 are vectorized. The first vector is called vector 1 and the last one vector 2. If there still remain vectors between vector 1 and vector 2 other than one unit vector, the whole process is repeated again, starting with the adding of pixels to the object, rather than background.

After this any junction will be ideal.

When we have reached this situation we still have to determine the position, orientation and the size of the angle. This is an easy task. The position is the position of the D-neighbour (or direct neighbour) of the start point of vector 2, which belongs to the background, the size of the angle is the difference between the direction of the vectors, and the orientation can be determined by examining the neighbourhood of the newly found junction pixel. If the angle is less than 60 degrees, the orientation of the angle is determined by the orientation of the background D-neighbour of the junction pixel. If the angle is greater than 60 degrees the orientation is formed by the background I-neighbour (or indirect neighbour) of the junction pixels. The orientation is expressed in Freeman codes, and will be called the junction direction

We give the algorithm introduced by the above ideas - in three parts:

1) A recursive algorithm to delete intermediate pixels and recalculate vector1 and vector2;
2) A recursive algorithm to add the intermediate pixels; and,
3) An algorithm to prescribe the intermediate pixels and to store the found junction pixel;

as follows:


```
PROCEDURE DELETE_INTERMEDIATE_PIXELS
INPUT :         Pointer chain of contour pixels;
                        Position in chain;
                        Vector1,vector2;
OUTPUT :        Pointer chain of contour pixels;
                        Vector1,Vector2;
BEGIN
                REPEAT
                        go on one pixel in chain;
                        IF pixel is not twice in the contour THEN
                                BEGIN
                                delete pixel;
                                store coordinates in buffer;
                                END
                        ELSE restore deleted pixels;
                UNTIL startpoint of vector2 is reached;
                put new contour pixels in contour chain;
                go back in the pixel chain until first pixel of vector1
                is reached; vectorize pixel chain until end point of
                vector2 is reached;
                vector1 := first found vector;
                vector2 := last found vector;
                IF 8-distance between endpoint of vector1 and start point vector2 > 1
                then
                delete_intermediate_pixels(. . );
                remove stored pixels from buffer;
END;
```

```
PROCEDURE ADD_INTERMEDIATE_PIXELS
Input :          Pointer chain of contour pixels;
                        Position in chain;
                        Vector1,vector2;
Output :         Pointer chain of contour pixels;
                        Vector1,Vector2;
BEGIN
                 REPEAT
                        go on one pixel in chain;
                        add neighbours;
                 UNTIL startpoint of vector2 is reached;
                 put new contour pixels in contour chain;
                 go back in the pixel chain to first pixel of vector1 is reached;
                 vectorize pixel chain until end point of vector2 is reached;
                 vector1 := first found vector;
                 vector2 := last found vector;
                 IF 8-distance between endpoint of vector1 and start point vector2 > 1
                 THEN
                 add_intermediate_pixels(. . );
END;

PROCEDURE DELETE_NOISE_AT_JUNCTIONS
Input: Pointer chain of contour pixels;
                 Pointer chain of objects * ;
Output: Coordinates of junction pixels;

Vectorize contour; Put vectors in a chain;
Start at the endpoint of first vector of the first object1;

REPEAT
        REPEAT
                IF length of preceding vector exceeds limit1 THEN
                BEGIN
                vector := vector1
                length := 0;
                REPEAT
                        Go to next endpoint;
                        length := length + length preceding vector;
                UNTIL length exceeds limit2 OR
                        length preceding vector exceeds limit1;
                IF length preceding vector exceeds limit1 THEN
                        BEGIN
                move forward in pixel chain until end point of vector1 is
   reached;
                IF directions of intermediate vectors are between
                directions of vector1 and vector2 THEN
                        delete_intermediate_pixels(. . );
                ELSE
                        add_intermediate_pixels(. . );
        END;
```

* In this case we mean with an object a group of connected pixels

C5.6 THE TON (THINNING-OF-NODES) ALGORITHM

The reader is reminded that this algorithm has been developed to replace (see C5) the existing thinning algorithms which create node dislacement (C5.2).

The algorithm starts with repairing the junctions (as described in section C5.5) and identifying junction pixels. We chose to insert junction processing in the thinning algorithm of Pavlidis [PAVLIDIS,1980] modified by Arcelli [ARCELLI, 1981].

After a thinning iteration the junctions are processed. If the angle of a junction is smaller than 63.4 degrees (from now on we will call this the 45 degrees case) then both the 'x' marked pixels of FIGURE 28 are deleted. If the angle is bigger than 63.4 degrees (from now on called the 90 degree case), only the 'x' pixel furthest from the junction is deleted.

Then the junction pixels have to be replaced. In the 45 degree case this involves replacing the pixel in the position of the rightmost 'x' marked pixel of FIGURE 28. In the 90 degree case this newly deleted pixel becomes the junction pixel.

When this is done a new thinning iteration is performed, after which the same processing of junctions is repeated. This process goes on until one of the branches becomes skeletal.

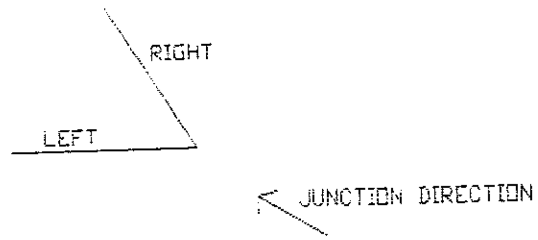
Then we have to make a distinction between separate situations, which have to be dealt with differently. FIGURES 36 a - c depict these situations after the thinning iteration in which one of the branches was skeletal.

FIGURE 36 AFTER ONE OR MORE THINNING PASSES IN THE TON ALGORITHM - ONE OF THE BRANCHES HAS BEEN PROCESSED WHILE IT WAS SKELETAL

```
....x....        .1.....        .x.......
....x....        ..1....        ..x......
..jyxyj..        ..j1...        .j2y2....
111111111        ...y,..        111111111
.........        xxxxxxx        .........

    a               b               c

where:

x,y : non skeletal pixels.
1,2 : skeletal pixels.
j   : junction pixel.
. , : background pixel. (value = 0)
```

The three cases (a,b,c) shown in FIGURE 36 are discussed.

To facilitate discussion we shall refer to the right or left branch as seen from the junction direction.



Case a.

The 90 degree case: one of the branches is skeletal. In this case the pixels marked y in fig 36a have to be deleted. There are two of these pixels in the figure, because there are two junction pixels. Since it is impossible to get one configuration by rotating the other, each of these two junctions pixels represent a different situation.

Case b.

The 45 degree case: The diagonal branch is skeletal, the horizontal is not.

In this case the y-marked pixel in FIGURE 36b can be deleted and the ',' marked background pixel converted to an object pixel. With this we run the risk of incorrectly extending the skeleton, such as when the horizontal branch does not go on after the junction. To avoid this, we check whether the ',' marked pixel was flagged for deletion in the previous iteration of the thinning algorithm. If it was, we declare it skeletal, if not we remove the junction pixel from the chain, so it will not be further processed.

Case c.

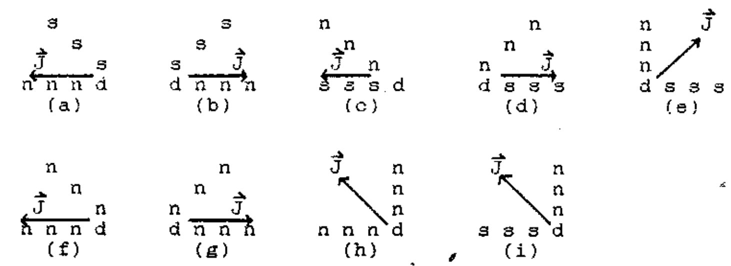
The 45 degree case: The horizontal branch is skeletal, the diagonal is not.

In this case the leftmost '2' pixel in FIGURE 36c has to be deleted. This configuration has a skeletal left branch. Actually, the rightmost pixel marked 2 also should be deleted, but as there is no straightforward way to determine its position we will not consider this pixel. This means a skeleton of non unit thickness will emerge. Depending on the effect of subsequent processing, either a skeleton without displacement, or one with a displacement of one pixel will arise.

For the identification of the pixels we can make use of the position of the junction pixel and of the junction direction. Starting from the junction pixel we have to apply a Freeman code to get the position of one of the pixels to be deleted. If there are more than one another Freeman code has to be applied. From Figure 36 we can see that the Freeman codes which have to be applied depends on which junction (a - c) we are

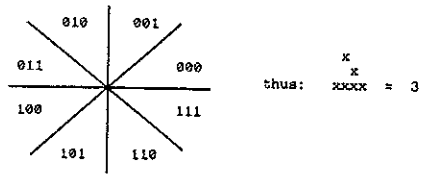
processing. Furthermore the shown configurations (with the exception of a) also have their mirrored counterparts which will have to be dealt with separately. Of course the direction of the junction also prescribes the Freeman codes to be used. However if we prescribe the Freeman codes relative to the junction direction the process becomes rotation invariant.

If we do this we will have to deal with 5 cases separately. Together with the configuration without a skeletal branch, this gives the 9 different cases depicted below:



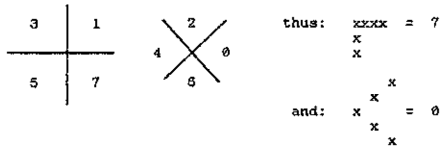
To determine the position of, for example, the pixel d we will have to add to the junction direction 3 in the cases (a), (e), and (f), direction 5 in the cases (b), (d), and (g) and direction 4 in the cases (e), (h), and (i). However in the case (a) the pixels to be deleted are different from those in (e) and (f) - because the configuration skeletal / non-seletal is different. The same also follows for the cases (b), (d), and (g) and the case (e), (h), and (i). Thus in every case we have to add different values to the direction of the junction pixel to find the pixel to be deleted.

FIGURE 37. POSSIBLE JUNCTION DIRECTIONS USING 3-BIT CODES



a:

Possible 45 degree angle configurations and their variable values.



b:

Possible 90 degree angle configurations and their variable values.

Together with the configurations without a skeletal branch, we have 9 different groups in which the pixels to be deleted can be determined by starting from the junction pixel, and applying a fixed Freeman code relative to the direction of the junction.

The same method can be applied for determining which branch is skeletal. For this we must look at the neighbour of the junction pixel which is as distant as possible from the other branch.

If this pixel is skeletal then the branch is skeletal, if not, then the branch is not skeletal. In TABLE 6 below these pixels are called branch pixels. TABLE 6 gives a

summary in which the addition value represents the difference in value between the junction direction and the Freeman code of the pixel to be determined.

TABLE 6 THE DIFFERENCE BETWEEN THE JUNCTION DIRECTION AND THE FREEMAN CODE

Junction Angle	Direction code of the junction	Addition value for branch pixel *	Skeletal Branch	Addition value of the pixel $ to be deleted
45 degrees	0,2,4,6			
		2,7	none	5,4
		2,7	left	5,7
		2,7'	right	4
45 degrees	1,3,5,7			
		2,7	none	3,4
		2',7	left	4
		2,7'	right	3,4
90 degrees	0,1,2,3, 4,5,6,7			
		2,6	none	4
		2',6	left	5
		2,6'	right	3

* The skeletal pixel is marked with a '.
$ When two values are given, two pixels have to be deleted. We find the first one by calculating the Freeman code from the junction direction and the first value and applying this on the position of the junction pixel. The second one is found by calculating the Freeman code using the second value and applying it on the position of the first found pixel.

The position of the first pixel also presents the new position of the junction pixel. This, together with the junction angle of the configuration determines in which group the configuration falls. This group then determines the pixels which will have to be processed to avoid node displacement.

When the deletion is done we still have to replace the junction pixel. From FIGURE 36 one can see that the new position is always the position of (one of) the deleted pixel(s). That is why this operation does not pose any problems and is not elaborated any further.

In this way it is made possible to decrease Y-displacement.

Let us now look at the problem of T-displacement. T-displacement occurs when only one of the branches is skeletal. This means we only have to prevent it in these situations. As has been said, T-displacement can be prevented by declaring pixels to be at the 'other side' of the skeletal branch. The pixel nearest to the junction pixel can be determined as we did while preventing y-displacement. This also holds for determining the initial direction in which we should follow the contour.

The only problem now is that we do not know where to stop. In most cases this problem is not difficult to solve; we just go on declaring pixels of the contour to be skeletal until another skeletal pixel is found. This means we have reached the other side of the node region. However if the skeletal branch is thicker at the other side of the junction this method will not work, because then the end of the node region is not marked by a skeletal pixel. We can prevent this by establishing a threshold value denoting the maximum number of pixels which can be declared skeletal. If after reaching this value no already skeletal pixel is found all pixels which have been declared skeletal are turned to their original value again. It is quite obvious that this threshold value should be equal to the maximum line thickness of the image and be decreased with every thinning iteration with two pixels. However, when there are junctions in the image which are connected by a branch and this branch has a length which does not exceed this threshold value problems still arise. Figure 38a demonstrates this situation.

FIGURE 38a. PREVENTING T-DISPLACEMENT: ALL THE PIXELS MARKED Y ARE WRONGLY DECLARED SKELETAL IF THE THRESHOLD VALUE IS HIGHER THAN 6 PIXELS

```
xxxx                      ssssss
 xxxx            sssssssss
   xxxxx         ssssssss
         xxxssss
        xxxy
       xxxy
        xxxy                   s
         xxxy        sssss
          xxxyssssss
           xxxx
            xxxx
```

We can prevent this by determining next to which kind of junction the found pixel is. If this junction forms an acceptable combination with the junction from which the process was started, then the pixels are declared skeletal. Figure 38b gives examples of acceptable combinations of junctions (the configurations also stand for their rotated or mirrored versions).

FIGURE 38b. ACCEPTABLE JUNCTION COMBINATIONS FOR PREVENTING T-DISPLACEMENT

```
.....xxxx....     .xxxx..xxxx....     ..xxxx.....xxxx.
.....xxxx....     ..xxxx.xxxx....     ...xxxx...xxxx..
11111xxxx1111     ...xxxxxxxx....     ....xxxx.xxxx...
.............     111xxxxxxxx1111     111xxxxxxxx11111
```

An extensive investigation of the junction combinations shows that if one starts from a junction with code J, then finding a junction with code J or J + 1 means that no pixels should be declared skeletal.

A less verbal description of this algorithm for preventing T-displacement follows, along with a procedure for preventing node displacement (i.e. Y-displacement)

```
PROCEDURE PREVENT_T_DISPLACEMENT;

INPUT :.    junction pixel record:
                        row;
                        column;
                        junction angle;
                        junction direction.
                image : matrix (n-row x n-column) of integer values;
                threshold;
OUTPUT :  image : matrix (n-row x n-column) of integer values;

            BEGIN
            flag junction pixels in image;
            examine neighbours of current_pixel;
            calculate position of first pixel at the other side of skeleton;
            calculate direction in which contour must be followed;
            i := 0;
            pixel_found := calculated pixel;
            WHILE pixel_found is not skeletal AND i < threshold
                AND other_junction_found := FALSE THEN
                BEGIN
                i := i + 1;
                examine neighbours;
                if neighbour := contour_pixel THEN
                    IF junction direction := unacceptable THEN
                        other_junction_found := TRUE;
                store position;
                declare pixel skeletal;
                find next contour_pixel;
                END;
            IF other_junction = TRUE OR i = threshold THEN
                turn pixels with stored position to original
                values again;
            END.

PROCEDURE PREVENT_NODE_DISPLACEMENT;

INPUT :.    image : matrix (n-row x n-column) of integer values;
            chain of junction pixel records:
                        row;
                        column;
                        junction angle;
                        junction direction;
OUTPUT :  image': matrix (n-row,n-column) of integer values;
```

```
BEGIN
current_pixel := first junction pixel;
examine neighbours;
IF all branches are skeletal remove junction pixel from chain;
IF no branch is skeletal THEN
     BEGIN
     calculate position of to be deleted pixels;
     delete calculated pixels;
     END;
IF one branch is skeletal THEN
     BEGIN
     calculate position of to be deleted pixels;
     IF calculated pixel is a background pixel THEN
          remove pixel from the chain;
     ELSE
          BEGIN
          delete calculated pixels;
          prevent_T_displacement(current_pixel,chain of
               junction pixel records,
                              image, threshold,image);
          END;
     END;
exit;
current pixel := next junction_pixel;
END.
```

Now we are able to give the algorithm which thins the image while diminishing node displacement:

```
PROCEDURE THIN;

INPUT : image: matrix (n-row,n-column) of integer values;
OUTPUT: image: matrix (n-row,n-column) of integer values;

     BEGIN
     follow contour and put it in a chain;
     delete_noise_at_contours(pointer chain of
                              contour_pixels,
                              chain of objects,
                    c         hain of junction pixels);
     REPEAT
          thin_image;
          prevent_node displacement(image, chain_of
                              junction_pixel records,
                              image);
     UNTIL image is skeletal;
```

In concluding this discussion of the TON algorithm, TON has been implemented in Pascal on a DEC VAX 3600 mini computer, with promising results. However, only artificially made images were processed. This means that no predictions can be made about the percentage of junctions which are correctly processed by TON. However, once a junction has been identified it will be thinned according to the standards outlined above.

C6. LINE TRACKING WITH NODE EXTRACTION AND SEGMENT SPLITTING

Assuming that a proper skeleton is constructed, then actual vectorization can take place. The method which will give the most compact, and pleasing results makes use of B-splines. Several studies make use of the B-spline representation of line drawings [PAVLIDIS,1982;ROSENFELDetal.,1982]butthemathematicalcomplexityhasprevented us from concentrating on them.

Among them is a method [PARKER, 1988] which commences by identifying a start pixel. From this pixel the neighbours are examined and a vector is made from the start pixel to every neighbour. From these neighbours the process is continued with another neighbour of these neighbours, etc. In the meantime, for each constructed vector, the distance of all the found neighbours to the vector is computed, summed and seen as a measure of 'gap'. When a newly considered pixel yields a 'gap' which surpasses a certain threshold the pixel is considered as not part of the vector. When there are no other neighbours of the last accepted pixel the vector is considered as terminated at this pixel and a search is started for a new start pixel.

The problem with this method is that the identification of start pixels is not elaborated. Furthermore no objective criterion is given for defining the maximum acceptable 'gap. A third drawback is that the criterion to end a vector is not dependent on the relative position of the pixel being considered, but on the pixel already attached.

The first and the third problem are avoided in an approach [LANDY et al., 1985] which was originally developed by Ramer [RAMER, 1972]. This approach uses a recursive process which starts with a categorization of pixels in the thinned image. Among others, branch points (nodes) and end points (in the following they are both called branch points) are distinguished. The algorithm then looks for two end points which are connected. Then the connecting pixel is sought which has the greatest distance to the straight line between the branch points. If this distance exceeds a certain threshold value then the found point is labelled a cut point, and the whole procedure is repeated, but now with the cut point also as an end point. Consequently two distances to two straight lines are calculated. This 'dividing up' goes on until all the distances are smaller than the threshold value. Then the chain is considered to be vectorized and two new end points are taken to vectorize another chain. These two approaches are shown in FIGURE 39.

FIGURE 39. VECTORIZING METHODS

- Parker's method.

Σ ∂ < threshold;

-Ramer's method:

-Start at node pixel;
-End at node pixel;
-Find pixel with greatest distance to connecting line;
-Repeat process with three points;
-Until maximum distance is under threshhold;

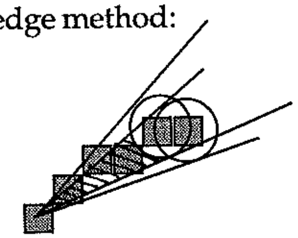
It is clear that this method is quite complicated, and needs repeated examination of the connecting pixels.

Still another method is used by Domogalla, [DOMOGALLA, 1984]. It is referred to as the wedge method. It uses an error circle of a certain size. The wedge is formed by the tangents to the circle constructed around the last accepted pixel, which intersect the starting point of the vector. The criterion for acceptance is that the wedge of the accepted pixel must intersect with the wedge of already accepted pixels (see also [Smith, 1987]) - see FIGURE 40.

The method presented here gives a vectorized representation of a raster image in which the amount of error equals that of the original raster image. That is, no error is added during the line extraction phase. It differs from the Ramer method, in that it does not apply a threshold. A consequence of this is that the amount of data reduction is relatively small. As the lines on a map are mainly made up of rather long straight line segments, considerable data reduction may be achieved. On the other hand, all the detail of the raster image is preserved. Furthermore, the method is quite straightforward in the sense that it just follows the skeleton and decides on the basis of the form of the part already vectorized, whether the next pixel is acceptable or not.

FIGURE 40. THE WEDGE METHOD FOR VECTORIZING

Wedge method:



-Wedges of all accepted pixels should intersect.

C6.1 A PROPOSED METHOD OF LINE EXTRACTION ADDING NO FURTHER ERROR

The method has a skeletonized image as input. In this the node and the end pixels are identified, making use of the crossing number (CN), which is the number of background to object conversions in the neighbourhood. Pixels with CN 1 are flagged as endpoints and with CN of 3 or 4 as node pixels. The line extraction starts at such a (CN 1, CN 3, or CN 4) point. It takes a neighbour and calculates the region in which the next pixel of the skeleton should lie to be considered lying on the straight line segment formed by the first two pixels. The method used to find the neighbour is questionable and is therefore considered below, under 'Organization of the Vectorization Process for the Whole Image'. When this pixel satisfies this criterion, the new acceptability region is calculated and the next pixel is examined to see if it is lying within this region. The process goes on until a pixel lies outside the acceptability region or if the last accepted pixel is a node.

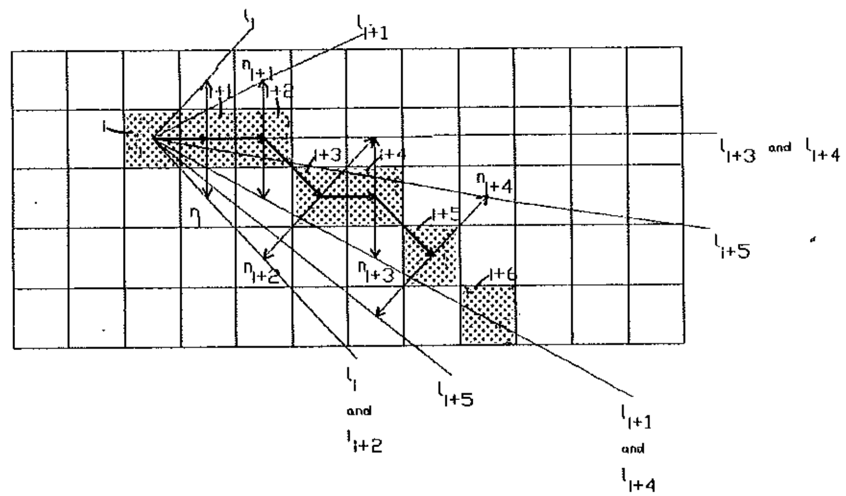
Then, the vector connecting the first found and the last found pixel is considered to be the vector representation of the processed skeleton segment. For defining how the acceptability regions should be computed, we should look more closely at the representation of a straight line in raster format. When we do this it becomes clear that the two Freeman codes pointing to, and pointing from the pixel do not say anything about the straightness of the line of which it is part. A perfectly horizontal line is represented by a row of pixels for which the above described Freeman codes are constant. But, if we rotate this line through a minimum angle then the representation of the line becomes two rows of pixels, with pixels getting a Freeman code of values 1 more or 1 less (depending on how we rotated the line) than other pixels. Here the error introduced by the scanning process becomes apparent. The true medial line of the scanned line is represented by pixels lying a given distance from this line. This given distance (really the scanning error) is used as the maximum separation for a pixel to be considered as a part of a straight line.

When we start vectorizing from one pixel, the acceptability region of the second pixel is 360 degrees, so the second pixel is always accepted. If this second pixel has a Freeman code (in the direction towards a third pixel) which is one more, equals, or is one less than the Freeman code from the starting pixel to the second pixel, then the third pixel is accepted. In the case where the Freeman code from the first to the second pixel is the same as that from the second to the third then the limits to be used when considering the fourth pixel are again that the Freeman code be one more, equal to or one less than that already used. But, should the Freeman code between the second and third pixel be one different from that betwen the first and second, then the limits to be used when considering the fourth pixel should be shifted by the same amount and in the same direction. Following this procedure subsequently accepted pixels should form a line.

The above described limits can easily be computed by taking the perpendicular of the Freeman code of the pixel, preceding the last pixel which was accepted to the line. This means this Freeman code is increased or decreased by 2. These normals are used as Freeman codes to find two neighbouring pixels (which will always be background pixels) of the last accepted pixel. The line connecting the starting pixels and the pixel which is found by applying the increased Freeman code forms the upper limit, while the line connecting the starting pixel with the pixel found by applying the decreased Freeman code forms the lower limit.

The acceptability criterion thus is that the line connecting the pixel to be accepted and the starting pixel should lie in between the two limits. That is, as all the lines go through the starting pixel, the tangent of this line should lie within the tangents of the upper and the lower limit. With this however, the shape of the line already formed is not taken into account. Wherever we are in the skeleton the acceptability criterion for the next pixel is the same. Therefore, we should also limit the domain of the limits by stating that a newly established limit should not lie outside any previous defined limit. In this way also, the shape of the already formed line is taken into account. Figure 41 demonstrates the whole procedure.

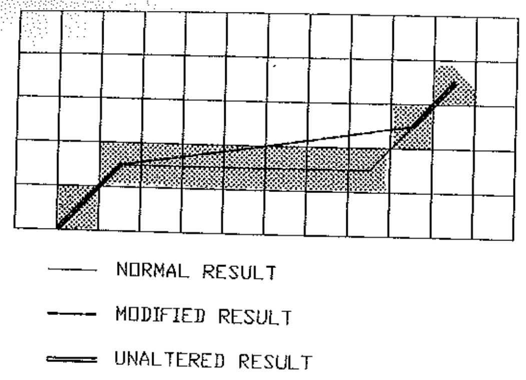
FIGURE 41 DEFINING THE ACCEPTABILITY CRITERION FOR ADDING A PIXEL TO A LINE



n_i normal used for calculating the limits (l_i)

l_i limits for acceptance of pixel i

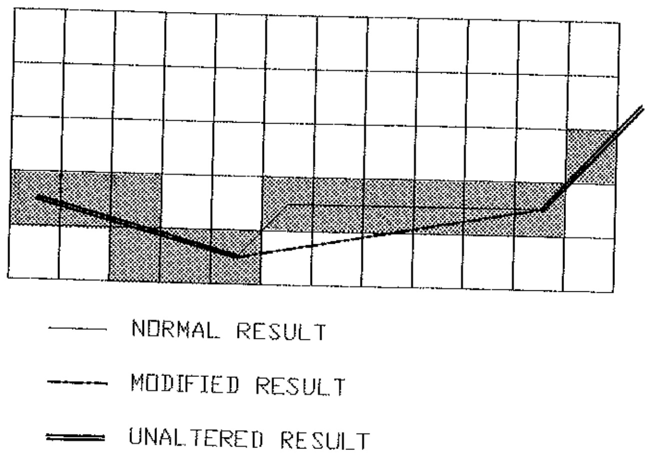
FIGURE 42 DETERMINING THE END POINT OF A VECTOR



Two refinements still have to be made. Suppose the algorithm has processed a branch which is perfectly straight and the last accepted pixel has a Freeman code different from the Freeman code of the already accepted pixels. The last accepted pixel should be considered as belonging to the next vector. Therefore, during the vectorization not only the Freeman code of the pixel which lies before the last accepted pixel is kept (the one we use for computing the limits) but also the Freeman code of the one before that. If these Freeman codes have an absolute difference of 1 then the last accepted pixel was accepted unjustly and instead of this, the pixel before this one should be considered as the end of the vector.

A similar refinement is made at the start of a new vector. When a branch starts with two pixels which have different Freeman codes it is assumed that the first pixel belongs to the preceding vector, than to the one under construction (see FIGURE 43).

FIGURE 43. DETERMINING THE START POINT OF A VECTOR



C6.2 AN ALGORITHM FOR VECTORIZING A PART OF A SKELETON BETWEEN TWO NODE PIXELS

PROCEDURE PROCESS_BRANCH; INPUT: node pixel coordinates;
OUTPUT: list of vector coordinates;

```
BEGIN      REPEAT
           current_pixel := node pixel;
           start_row := row;
           start_column := column;
           store coordinates in file;
           find neighbour;
           current_pixel := found pixel;
           IF Freeman codes differ THEN
                BEGIN
                store coordinates in file;
                go to exit;
                END
           calculate_limits;
           calculate tangent;
           examine neighbours;
                WHILE tangent is inside limits DO
                     BEGIN
                     calculate_limits;
                     take_inner_limits;
                     examine_neighbours;
                     END;
                     IF last pixel is accepted unjustly THEN
                          take previous pixel as end point;
                     store last accepted pixel coords in file;
       UNTIL amount of neighbours > 3;
       exit;
END;
```

C6.2.1 ORGANISATION OF THE VECTORIZATION PROCESS

Until now we only have considered the vectorization of a part of the skeleton, which lies in between two node pixels. When after vectorizing a part, we arrive at a node pixel problems arise. There is no objective choice possible about the branch with which we should continue the process. Two different methods can be applied.

Firstly, we can apply a contour follower on the skeleton and in the meanwhile just process the pixels. However, as every pixel in a skeleton appears in two different places in the same contour (or, in case of an island or a lake, is part of two different contours) we cannot be sure anymore in which way the contour is followed. This should not be too big a problem if we could be sure that every pixel indeed appeared twice in the set of contours describing the image. Of this, however we are not sure. In the case of an island or a lake it only appears once. This means that the contour follower should be altered so that it is sure that every pixel is processed twice. As this is done, then the result of the vectorization will be that every branch of the skeleton has been vectorized twice. Apart from the drawback of increased storage space and increased processing times, another problem arises. The proposed vectorization method has the property that it is not direction invariant. That is, if a skeleton is processed in one direction other vectors will be obtained than from processing the skeleton in the other direction. Consequently, the two vectors describing the same branch segment, will often not be the same. This means that additional processing will have to be done, in which vectors which are nearly the same, are made the same, or another vectorizing approach should be used.

The second method which can be applied, ensures that when arriving at a node a variable attached to the node is first examined. In this variable the number of branches is stored. If this is empty it examines the neighbourhood pixel and stores the number of branches evolving from it in the variable. It then decreases the contents of the variable by one and processes one branch. When it arrives at another node pixel this process is repeated. When it arrives at an end pixel it goes back to the last processed node and if the variable is not yet zero it processes one of the not yet processed branches. When it is zero it goes back one node pixel more. This process continues until the variable attached to all node pixels has in all cases the value zero.

One problem needs further elaboration. At nodes it is possible that a pixel has more than 2 neighbours (see FIGURE 44).

FIGURE 44. A PIXEL MAY HAVE MORE THAN TWO NEIGHBOURS

```
x   x
 x x
  xx
  x
 x
x
```

In FIGURE 44, apart from the problem that the true node position is not represented by a pixel, there is the problem that there exists a choice as to which neighbour to process next. To ensure the node is not missed, we thus always have to check to see if there is a neighbour which is a node. It is probably better while flagging the nodes, to also flag the neighbours of these nodes - as the neighbour of a node.

We can construct a record for each node of the following type:

```
#1: row;
#2: column;
#3: number of branches
#4: pointer to preceding node record;
```

and can implement the following algorithm:

```
PROCEDURE vectorize_image INPUT : image containing skeleton : matrix (row x column);
OUTPUT : list of vector coordinates;

BEGIN REPEAT    REPEAT
            scan image;
        UNTIL node found;
```



```
        current_node := found node;
        first_node := current_node;
        REPEAT
            WHILE #3 <> 0 do
                BEGIN
                IF #3 is empty then
                    BEGIN
                    examine neighbours;
                    fill node record;
                    END;
                #3 = #3 - 1;
                process branch;
                current_node := found node;
                WHILE #3 = 0 DO
                    current_node := preceding node;
        UNTIL current_node = first_node AND #3 = 0;
    UNTIL all pixels are scanned; END;
```

C7. SEGMENT MERGING

After line extraction the raster image is converted into a list of vectors with one or more attributes attached to it. However, compared to the original image some of the topology of the original paper image may be destroyed. Two particular cases can be distinguished.

One of these is a hardware problem. It happens when the internal memory of the computer is too small to contain the whole image. The image then is divided up in patches and every patch is processed, one at a time. Algorithms exist for this job and will not be elaborated any further here. When a patch is thinned then the ends of the cut branches will be seen as part of the contour and will be subject to deletion from the first pass. They will be treated as normal dead end branches. This means that when the image is skeletonized not only the width of the branch has shrunk, but also the length. The size of this decrease in length is approximately half the original line width. As the same has happened with continuation of the branch in the other image, there exits a gap of approximately the line width of the cut branch. It is important that this gap is closed and a relation is established between the segment in one image and the continuation of it in the other.

The other case is a matter of semiology. When the human eye looks at a broken line the brain merges this into one symbol. If we want to have the possibility to do operations on the whole symbol, it is thus necessary to develop a method which merges the different dashes into one segment.

Although, strictly speaking, the problem of undershoots is not a consequence of destroyed topology but of inaccuracies of the original image, it also will be dealt with here. The reason for this is that the problem is very similar to that of attaching the end of a broken line to a 'neighbouring' segment. Therefore we assume that the method presented below for solving the problem of connecting a broken line also applies to the undershoots.

C7.1 MERGING OF SEGMENTS ACROSS THE PATCH BORDERS

There are two different groups of methods to asses this problem. One can recreate topology in a raster phase or one can do this in a vector phase. Both methods and their consequences are described below.

C7.1.1 RASTER PROCESSING FOR CROSS PATCH SEGMENT MERGING

The first method deals with the raster image. When the image is skeletonized, the skeletons of branches with an endpoint near the border are extended someway or another. The only information about the branch we have at this stage is the Freeman codes of the pixels in the branch. This means the only possibility we have is to extend the skeleton in a direction prescribed by the Freeman code of the pixel which is neighbour of the end pixel. FIGURE 45 shows, however that this does not fully guarantee no gap between the branches at both sides of the patch border.

FIGURE 45. EXTENDING A SKELETON WITH ERRONEOUS RESULTS

```
. x x x s x x x . . . . . . .
. x x x s x x x . . . . . . .
. x x x s x x x . . . . . . .
. . x x e x x x x . . . . . .
. . . x e x x x x x . . . . .
. . . . e x x x x x x . . . .
    patch border                   x is an object pixel
. . . . . x x x x x x x . . .      s is a skeleton pixel
. . . . . . x x x x x e x . .      e is an extension pixel
. . . . . . . x x x e x x x .      . is a background pixel
. . . . . . x x x e x x x . .
. . . . . x x x s x x x . . .
. . . . x x x s x x x . . . .
```

This means that the result of the extension in the two patches should be checked. If the result is not satisfactory, the two original patches should, one at a time, be loaded into the memory again, and the extension should be repeated with a different sequence of Freeman codes, and checked again.

Another approach is to first compute the position of the pixel on the border to which the branch should be extended. According to this method first the end pixels of the skeleton, which are near the border of a patch are detected. Depending on the side to which the second patch should be 'attached', only the pixels near the upper-, lower-, left-, or right borders respectively are considered. When such a pixel is near the upper or the lower border, only the column number is kept. When it is near the left or right border only the row number is kept. When we do this also with the second patch, we have two sets of column or row numbers at our disposal. From these sets , pairs are derived, whose numbers show the least difference and whose difference is under a certain threshold. From these pairs the mean value is taken, which is rounded to the nearest integer value. This integer value respectively represents the row or column number to which the skeleton should be extended. The most obvious way to do this is to rasterize the vector between the end point and the found border point. Then some kind of pointer should be connected to the computed end pixel, which points to the neighbouring pixel in the other patch. When the skelton is vectorized, the pointer can be replaced by a vector.

FIGURE 46. EXTENDING A SKELTON WITH DISTORTED RESULTS

```
. . . . . . . x x s x x . . . . .
. . . . . . . . x x s x x . . . .
. . . . . . . . . e x x x x . . .
     patch border                      x is an object pixel
. . . . . . . . x e x x x . . . .      s is a skeleton pixel
. . . . . . . x e x x x . . . . .      e is an extension pixel
. . . . . . x x e x x . . . . . .      . is a background pixel
. . . . . x x s x x . . . . . . .
```

Both of these methods are very cumbersome, and both have the drawback that we are not able to control the form of the extension in relation to the form of the part from which and to which it is extended. Figure 46 presents an example of a somewhat distorted result.

C7.1.2 VECTOR PROCESSING FOR CROSS PATCH SEGMENT MERGING

The second method operates on the vectorized image. As in this phase, a considerable data reduction has taken place already, storage of the whole image in the internal memory will not be a problem anymore. When all the vectors are stored at the same time we have to be sure, that the coordinate systems used are merged into one, while the border coordinates are known in the new system.

The algorithm starts with identifying the endpoints near the borders. For this it is very convenient to flag the points from which only one branch evolves as end points, during the vectorization process. From these points pairs are derived which have a minimum distance to each other, which is below a certain threshold. When these pairs are constructed, the two vectors attached to the points are rewritten as a function and the point of intersection is computed.

We now have the coordinates of the point of intersection and we replace the end point of the vector by the found intersection point. The only thing which has to be done is to adapt the data structure, so that the new topology becomes explicit.

One thing needs further elaboration. We stated before that the pairs of points should be derived on basis of their mutual distance. We indeed think that this criterion is necessary and sufficient for most cases. The only thing which has not been considered is when the endpoints of one patch do not have a one to one relationship with the end points on the neighbouring patch.

This situation can occur because of two things:

1) The end of an actual 'dead end branch' is positioned on or very near the border of a patch.
2) A node is positioned on or very near the border of a patch (see FIGURE 47).

In the first situation nothing should be done. In most cases this is very easy to verify by means of the distance criterion. However, failures can occur, when the end point is sufficiently close to an end point of another branch in the other patch (see FIGURE 47).

This nearby branch will always be a 'border crossing branch'. Then the 'dead end branch will be connected to a node constructed according to the method described below. As the number of nodes constructed on or very near to the border will be very low, manual editing to prevent erroneous nodes is not a very great burden. For this it is very helpful, if after the merging process, the constructed nodes are depicted on the screen, with the possibility of undoing the erroneous ones. The operator then can make a visual comparison of the given data with the original map. Deleting certain nodes means replacing the node, by its original end points. This means that the end point belonging to a constructed node should be kept until the operator has done his or her job.

When a node is positioned on or very near a patch border, it will have disappeared after the vectorization process. However, the distance criterion used also applies on this situation. This means that we should give the algorithm the possibility of attaching more that two vectors to each other. Extensive study of this situation did not show any cases in which this possibility should lead to erroneous results.

FIGURE 47. SPECIAL CONNECTIONS ACROSS A PATCH BORDER: RECONSTRUCTION OF A NODE

```
. . x x S x x . . . . . x x S x x .
. . . x x S x x . . . x x S x x . .
. . . . x x E x x . x x E x x . . .
. . . . . x x E x x x E x x . . . .
          patch border
. . . . . . x x E x E x x . . . . .
. . . . . . . x E E x x . . . . . .     . background pixel
. . . . . . x x S x x . . . . . . .     x object pixel
. . . . . x x S x x . . . . . . . .     S skeleton pixel
. . . . x x S x x . . . . . . . . .     E extended pixel
. . . x x S x x . . . . . . . . . .
```

C7.2 MERGING A BROKEN LINE INTO ONE SEGMENT

This operation is very similar to the one described above. Also here a choice can be made between raster and vector processing.

C7.2.1 MERGING A BROKEN LINE INTO ONE SEGMENT: RASTER PROCESSING

To detect broken lines in maps, one can employ the same method as has been described above. The problem of comparing two different patches does not exist in this case, however now problems of how far to extend the branch occur. Parker says about this problem of 'extending a line beyond a gap', with reference to Pavlidis et al. [PAVLIDIS et al., 1985], that 'this may turn out to be harder to solve' [PARKER, 1988, pp.77].

One can also make use of methods like the Hough transform [BALLARD, 1981] or dynamic programming [YAMADA, 1984]. These methods need some a priori knowledge about the curve to be detected. With features like colour, thickness etc. this might not be a very big problem. Knowledge about the shape however will need considerable study of the map which is processed. Furthermore, experiments show that using the Hough transform to detect straight linear feature, give results which are highly ambiguous (See Section C9).

C7.2.2 MERGING A BROKEN LINE INTO ONE SEGMENT: VECTOR PROCESSING

Methods which process the image after it has been vectorized include several expert system based approaches and syntactic methods. These methods also assume a priori knowledge about the feature to be detected [NAZIF et al., 1984; YOU et al., 1979]. Kasvand [KASVAND, 1986] has developed a method, which does not need such knowledge.

It is our opinion that the problem in maps is not so complicated that these complicated approaches are necessary. Linestyles on maps are restricted to a few simple types. Textures on German and Dutch topographic maps provide examples and we can see that the main structures used are very few. They are depicted in FIGURE 48.

FIGURE 48 DIFFERENT LINE TEXTURES APPEARING ON TOPOGRAPHIC MAPS
(S denotes a point symbol)

```
(a)   - - - - - - - - - - - - - - - - - - - - -
(b)   -- -- -- -- -- -- -- -- -- -- -- -- -
(c)   -- . -- . -- . -- . -- . -- . -- . --
(d)   . . . . . . . . . . . . . . . . . . .
(e)   |--| |--| |--| |--| |--| |--| |--| |-
(f)   -- S -- S -- S -- S -- S -- S -- S --
(g)   S S S S S S S S S S S S S S S S S S S
(h)   | | | | | | | | | | | | | | | | | | |
```

(S denotes a point symbol)

The main features of these textures are:

1) They consist of rather short vectors;
2) The number of vectors which make up a single texture element is rather limited;
3) A texture element contains at least 2 end points; and,
4) The texture elements make up a repetitive sequence, each sequence containing one or two texture elements.

The linestyles depicted in figure 48a-d are the one most widely used.

These observations lead to the following assumptions:

1) a broken line is characterized by segments, which contain at least two endpoints;
2) this selection can be reduced on basis of the length of the contained vectors (All vectors should have a length which is under a threshold L1) and the number of contained vectors (All objects which consist of at most N vectors);
3) if the element contains vectors which have a length above a certain threshold L2 (L2 << L1) we consider the element to be linear;
4) if a linear element contains only vectors of generally the same direction we consider it to be straight linear;
5) starting from a straight linear element we can find the next element by applying a direction criterion, either by applying the direction of the vector(s), or by applying its normal;
6) if the next element found is different from the one from which the process started, then the element following that next element should be equal to one of the two preceding elements;
7) to find the next element we can in all case make use of a distance criterion (next element should be within a distance D);

8) the occurence of length of the blank space between the symbols should have at least the same order of repetition as the elements; and,
9) to be characterized as a linear segment, the sequence must contain at least S repetitions.

With these assumptions we are able to detect a broken line in a vector image. For this it is convenient to establish the following data structure:

For each found element a record is constructed as follows:

Element record :
- #1 : Number of end points;
- #2 : pointer to first end point record;
- #3 : pointer to first vector record;
- #4 : pointer to first node record;

End point record :
- #1 : x coordinate;
- #2 : y coordinate;
- #3 : pointer to next node record;

Vector record:
- #1 : x coordinate start;
- #2 : y coordinate start;
- #3 : x coordinate end;
- #4 : y coordinate end;
- #5 : pointer to next vector record;

Node record:
- #1 : x coordinate;
- #2 : y coordinate;
- #3 : pointer to next node record;

Chain record:
- #1 : x coordinate first point;
- #2 : y coordinate first point;
- #3 : x coordinate second points;
- #4 : y coordinate second point;
- #5 : pointer to next chain record;
- #6 : pointer to previous chain record;

However, we have not considered the termination of these lines. We make the assumption that it should end at another linear feature or, if such a feature is absent, it should end at the last element. This linear feature should exist within the expected repetition distance and in a direction formed by the last two elements of the broken line. When such a feature is found, the broken line should be attached to this feature by means of a node.

When there is a node present at the expected place, we can simply take this node as the end point of the broken line. When it is absent, a search through the vector file is necessary to see if a vector crosses the region where the node is expected. For this widely accepted methods exist so it will not be elaborated any further at this point.

When the line is detected from its start point to its end point the representation of the line in the vector file should be replaced by a chain of vectors. For this we can take the found endpoints of the elements and the found or constructed nodes as coordinates in all cases except the case shown in FIGURES 48e and 48f. In the case of FIGURE 48e we can use the point which is the mean of the two end points. In the case of FIGURE 48f we take the nodes of the element as the coordinates for the new vector. The appearance of the line should be controlled by an attribute which is attached to the vector chain.

C7.2.3 AN ALGORITHM FOR MERGING A BROKEN LINE INTO ONE SEGMENT: VECTOR PROCESSING

The following algorithm gives a first approximation of the method outlined above; the repetition has not been taken into account. The algorithm starts with storing the groups of connected vectors which satisfy the criteria mentioned above in the records given above. It then continues processing these records. It takes the first record and looks if there is another element which satisfies the criteria. When this is successful, then the element record is deleted and a chain record is created. Then the process is repeated with the found element. This continues until no element can be found anymore. Then the last element is put in a chain record. Because it is possible that the first found element is in the middle of a broken line, the process is continued with the first processed element, with, if relevant, an inverted distance criterion. When this is also completed, the algorithm starts at the beginning of the remainder of the chain of element records and repeats the whole process again. When no element is found the element record is also removed, and the element is stored in a chain record. The list will then consist of one chain record, which makes it possible to identify isolated elements very easily. The algorithm stops when the list of element records is empty.

```
PROCEDURE BROKEN_LINE_DETECTION; INPUT : vectors with explicit topology;
thresholds;
OUTPUT : vectors with explicit topology;

    BEGIN
    REPEAT          read end points;
       read attached vectors and nodes;
       IF vector is connected to less then N other vectors AND
              lengths of all connected vector < L1 then
              make records;
  UNTIL all vectors have been read;
          REPEAT          start direction := 999;
       read element record;
       current element := element read;
       start element := current element;
       IF number of end points = 2 then
              compute direction between end points;
       IF number of end points = 4 THEN
              IF number of nodes = 2 THEN
                     compute direction between nodes;
       start direction = direction;
       FOR counter := 1 to 2 do
              BEGIN
```



```
start new list of chain records;
IF number of end points = 1 THEN
       REPEAT
              read other element record;
              next element := element read;
              IF next element satisfies distance
              criteria THEN
                     BEGIN
                     store element in chain record;
                     remove element record;
                     END;
              current element := next element;
       UNTIL a record has been accepted OR
              all records have been examined;
IF number of end points = 2 THEN
       BEGIN
       compute direction between end points;
       REPEAT
              read other element record;
              next element := element read;
              IF other element satisfies direction
                     and distance criteria THEN
                     BEGIN
                     store element in chain record;
                     remove element record;
                     END
              ELSE
                     BEGIN
                     direction := direction+90deg
                     if other element satisfies
                     direction and
                     distance criteria
                     THEN
                     BEGIN
                     store element in
                     chain record;
                     remove element record;
                     END;
                     END;
              current element := next element
       UNTIL a record has been accepted OR
              all records have been examined;
END;
IF number of end points = 4 THEN
       BEGIN
       IF number of nodes = 2 THEN
              BEGIN
              compute direction between nodes;
              REPEAT
                     read other element record;
                     next element := element read;
                     IF next element satisfies
```

```
                                        direction
                                        and distance
                                        criteria THEN
                                        BEGIN
                                        store element in
                                        chain record;
                                        remove element
                                        record;
                                        END;
                              current element := next
                                              element;
                              UNTIL a record has been
                              accepted OR
                              all records have been
                              examined;
                         END;
                    END;
               store current element in chain record;
               current element := start element;
               IF start direction <> 999 THEN
                  direction := (start direction + 180) MOD 360
          END;
          remove element record;
          attach the two last made lists of chain records to each
          other;
     UNTIL no element records are left;
        process list of chain records to create vectors;
END;
```


C8. TOPOLOGICAL RECONSTRUCTON

It can be seen from FIGURE 17 that this section must involve several activities:

1) arc attribute coding;
2) complex object construction; and,
3) complex object feature coding

Other activities are also implied and are addressed in this section as well. To code an arc, the arc must exist, and although we have acknowledged that arcs can be found in the collection of pixels between nodes the segmentation techniques found in digital image processing can also be used to construct lines which may be arcs; thus in this section segmentation will be considered. Also constructing objects may utilize symbol recognition techniques which are also dealt with in this section. The subject matter of this section will be dealt with under the following headings:

1) construction of arcs;
2) construction of complex objects and polygons;
3) automatic attribute coding;
4) complex object coding; and,
5) point symbols and text.

C8.1 CONSTRUCTION OF ARCS

In section C1 arcs were introduced as being a component of networks and domains. Graphically arcs are represented as line features, and topologically they connect nodes; thus we have already considered their construction when addressing skeletonization and skeletonizing with node improvement. However there are techniques which in digital image processing seek to find lines - these are segmentation techniques and are dealt with below.

C8.1.1 SEGMENTATION IN RASTER FORMAT

The problem in raster processing is that it is very difficult to segmentize an object. Thus it is only possible to look at objects of the lowest order (pixels) and of the highest order (groups of connected pixels). The objects we want to code are of some intermediate order. We want to distinguish a railway from a motor way even though they intersect. This means that segmentizing the image is a very important task in the process of attribute coding.

However, there are two methods of segmentation, which are already possible in the raster phase: a separation on basis of colour and on line thickness.

With the first method it is assumed that we do not scan the original map, but instead the colour separates which make up the map. Also it is assumed that the different images derived from the scans has been made fit perfectly together. Then, during a comparison of (patches of) the four different files, the binary values of the corresponding pixels are replaced by 'more' bits. The number of bits used depends on the number of colour separates. This means that a three colour map will need a four bit (including black) code per pixel, In this code black overrules all the other colours, which means that when the black bit is on, the object is considered to be black, whatever colour the other bits show.

Suppose that the maximum line thickness and the maximum point symbol thickness is less then n pixels. Then after thinning the new image in the integer part of 1/2n + 1 passes, all the lines and symbols will be skeletal, while the area symbols are not [ARCELLI et al., 1984].

It is our conviction that the basic shape properties of line and point symbols are best described by the skeleton of the image, area symbols are best described by the contour of the symbol.

We will leave processing of area symbols for this moment and go on with the processing of line and point symbols.

During the thinning process the pass number in which a certain pixel is declared skeletal is stored. This value gives an approximation of the width of the original line. Maxima in the gradient of these values represents a separation between two different linear features. To incorporate these points in the process the vectorization algorithm should be changed. The algorithm should break a vector not only at nodes but also at these pixels. This means that they should have been flagged accordingly.

C8.1.2 SEGMENTATION IN VECTOR FORMAT

The actual segmentation takes place during the vectorization of the skeleton. While vectorizing, the algorithm should break a vector at the following pixels:

1)a node pixel,
2)a pixel which is flagged as having maximum gradient.
3)a pixel with a colour code which is different from the already vectorized pixels.

However, not every line symbol on a map is represented by a normal line.

The greatest difficulties are posed by broken lines. It is possible that the sequence of elements making up the line symbol can be converted into a segment. For recognition purposes it is important to store the properties of the individual elements and attach them to the segment. On basis of these, recognition can take place quite easily.

Recognition of unbroken deviating line styles, once detected is also quite easy. Figure 49 gives examples of these lines. As can be seen they are made up of repeating symbol which contains a rather small number of vectors, nodes and endpoints. Detection will give the least problems when every line style is stored according to the basic properties of the repetition elementsuch as number of vectors, nodes and endpoint and the length of the vectors. This information should also contain the start- and endpoint of the vectors, so that the new segment can be computed.

FIGURE 49. EXAMPLES OF UNBROKEN DEVIATING LINE STYLES

With this a set of segments are obtained in which each segment consists of vectors which represent a line of one colour and one thickness. For each segment we also know the length of the consisting vectors and their relative angle. On these variables several statistical measures can be applied.

For the storage of the information we can make use of a record of the following type:

Segment record:
#1 : number of nodes(*);
#2 : pointer to first node(*) record;
#3 : pointer to first end point(*) record;
#4 : colour;
#5 : line thickness;
#6 : pointer to line style description;
#7 - #n : statistical measures;

(*) In the terminology used here a node is distinguished from an end point but includes vector ends because of a change in colour or line thickness.

C8.2 CONSTRUCTION OF COMPLEX OBJECTS AND POLYGONS

Arcs and nodes form networks and polygons. But point objects such as complex point symbols made from short arcs and (possibly) text, and complex linear features (e.g. patterned and parallel lines) all also appear on maps. Certain techniques have been developed to handle them, and these are considered here.

C8.2.1 OBJECT CONSTRUCTION

When we have segmentized the image we are able to construct higher order objects. If segments can be merged into a higher order object, they should be connected, that is, two subsequent vectors should have the same node and colour, they should have the same line thickness and have statistical measures in the same order of magnitude. We can score this in the following way:

Object record :
#1 : object identification;
#2 : pointer to first segment record:

Probably after this phase the class of object records with object identification "unknown" will be quite large. This group of unclassified objects will mainly consist of segments which are black and linear (that is, have a high total length measure) and objects which consist of segments which are non linear, e.g. point symbols and text.

C8.2.2 LINES AND PARALLEL LINES

The first group will enclose all the black, normal lines in the image. Most will constist of lines bordering linear features, like a road. This introduces the problem of how to detect a parallel line.

Nagao et al. have tried to solve, and claim to have succeeded in solving, this problem by operations on the raster image [NAGAO et al., 1988]. They apply a vertical and a horizontal scan and count sequences of black and white pixels to detect parallel lines.

Also a method suggested by Martinez-Perez et al. might be adapted to solve this problem [MARTINEZ-PEREZ et al., 1987]; they found the medial line of a raster line by considering only the contours of that line, and taking the middle of two opposite contours. However, then the problem of processing junctions re-occurred.

A third proposed method is to apply the Hough transform [BALLARD, 1981; DUDA, 1972; SLANSKY, 1987] on the skeleton. As experiments have shown also this solution is difficult (See section C9.3).

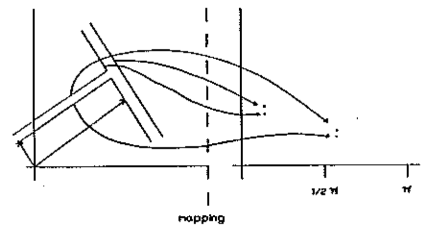
It is our conviction that this problem can best be solved when the lines are in vector form. We propose a method which might be called a Hough transform adapted for vectors. In this transform every vector is mapped in a feature space:

F-(alfa, R);

in which alfa denotes the slope of the normal of the vector and R its length to the origin. When two vectors are parallel, they should only differ in R, a difference which is equal to the distance between the lines. Thus the problem of finding parallel lines comes down to finding vectors in feature space, which only differ a little in R (see figure 50).

When this method is applied with all vectors, in the case of a coloured line bordered by two black lines, three points will be found in feature space. Of these only the middle one is important. We can thus delete the other two. When the space between the two black lines is white we only get two points next to each other in feature space. To get the center line of these lines we can take the mean of the two R values In image space this comes down to translating the vector over a distance R/2 . However, then the topology of the translated vectors will be destroyed. Therefore it is necessary that the endpoints should be recomputed. The alfa, the new R and the old topology of the translated vector are the parameters with which this can be done. This means that when a parallel line with a blank interior is found the topology of the vectors making up this line should be stored first. Then the new intersections between the previously connected vectors should be computed. As this is a problem of a merely mathematical nature we will not deal with it any further.

FIGURE 50. MAPPING OF VECTORS IN FEATURE SPACE



C8.2.3 GENERATING POLYGONS USING COMPONENT LABELLING

Component labelling identifies regions (polygons). The contour pixels of polygons found by component labelling are, of course, on the polygon boundaries. For example, component label 1 may indicate France and component label 2 Belgium. Processing of polygon boundaries will generate the topological relation adjacency:

> if on one side of a line (of contour pixels), the label 1 is found, then the line will become known as part of the boundary of France;
>
> if the label 2 is found on the right side, it will become known that the line also belongs to Belgium;

and finally,

> it will be described as the boundary between France and Belgium.

The only non-trivial task is finding the labels associated with the boundary. When we know we have found the label 2, we then deduce it is Belgium. But how can we find that it is label 2? We do this by looking at one pixel and asking for its label. The problem is: 'At which pixel must we look to obtain information'. This is not easy, but in 'good' cases, the pixel is given by the inner bisector of the angle formed by 3 consecutive contour pixels.

C8.3 AUTOMATIC ATTRIBUTE CODING

When digitizing a map in a semi automatic way, coordinates, pointed to by the operator, are registered by the computer. At the same time however information about the kind of feature which is being registered has been given to the computer. In this way a set of coordinates is derived with attached to it a code about the quality of the feature, the so called attribute data. We will call this process attribute coding.

In automatic digitizing only registering of coordinates takes place. Until now no registering of attribute data, which could be termed automatic, has been possible. This is reflected by the report of Waters et all on automatic datacapture (Waters et el, 1989). They give a description of the possibilities of the VTRAK system, developed by Laser-Scan. In this they say that "the heart of VTRAK is the interactive mode for setting all feature extraction parameters and for interactive (but semi automatic) extraction of features selected by the operator (Waters et al, 1989, pp.380). As they conclude by saying the VTRAK system is made to meet the needs of the 90's this seems to be a rather good indication of the state of the art in automatic feature coding.

However, in some fields it is reported that progress already was made 12 years ago. Denegre et al report that at the French topographic mapping organization I.G.N [DENEGRE et al., 1978] a package, perfected by the "Centre d'Electronique de l'Armement" is used for the automatic coding of contour lines with their altitude. However, as they speak of processing contour lines rather than detecting them, it is assumed that the original image, contains only contour lines.

When we assess the problem of automatic feature coding we have to start with the information embedded in the original raster image. However, only a very limited amount of this information is accessible for the computer. During the process of vectorizing a part of the information inaccessible during the raster phase is made accessible. On the other hand the conversion process also implies a certain information loss. This is why we will have to look at both the information formats. We will first deal with the possibilities of the raster format after which we will concentrate on the vector format.

The preceding approach may be called a bottom up approach. This will however, not be enough to detect every possible feature. There will exist feature which are not recognised, or their will be features of which the properties 'are somewhere in between' two classes. Within the frame work outlined above, it is impossible to solve these problems. To do this a top down aproach may yield results. One of these is the contextual approach. De Simone has developed a method to do this [DE SIMONE, 1986]. Besides parameters describing the shape of a feature he takes into account which feature should be next to another feature (a pavement and a street) or which feature should be connected to another feature (a river and a sea). He writes: "An essential feature of most land parcels is access to a road or pavement. Working outward from the road supershape(s) means that all edge links could be parcel frontages. (..)the program proceeds to search through the datastructure to define a suitable side and back to an area which might be a land parcel."[DE SIMONE, 1986 pp.89]. Two remarks have to be made. In the first place De Simone does not describe the implementation of his proposed method. In the second place, it should be pointed out that his approach is cultural variant. With this we mean that the contextual parameters valid in one country may fail to describe the situation in another country. He for instance mentions long narrow strips beside a railway, which most probably will be fenced areas. This may be true for the situation in Great Britain, but going by train from Amsterdam to Enschede one sees hardly any fenced area. Still, we think that this approach is the only one which is left after all the explicit information of the images have been processed.

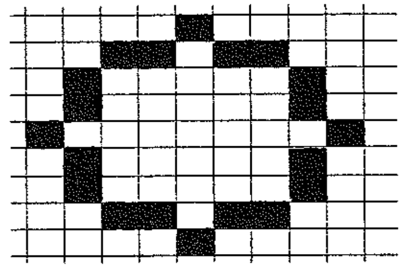
C8.3.1 AUTOMATIC ATTRIBUTE CODING IN RASTER IMAGES

In this stage the information we have at our disposal consists of pixel values and Freeman codes of contour pixels. As it is possible to extract vectors from the Freeman codes, automatic attribute coding may be performed by exploiting pixel values, Freeman codes or vectors extracted from these Freeman code. Each approach is dealt with below.

C8.3.1.1 AUTOMATIC ATTRIBUTE CODING USING PIXEL VALUES

Pixel values are used in the process of template matching. The feature, which has to be detected is depicted in a raster template (see FIGURE 51). The template is 'laid over the map' and the rate of similarity between the template pixels and the pixels on the map is computed. Then the template is shifted one row or column, and the process is repeated. The process stops when the whole map has been processed like this. When a local maximum in similarity occurs, which is above a certain threshold, the feature is considered to be detected. Illert has improved this method by applying a weight factor to each pixel of the template, which he derived from the success rate of each pixel in a previous recognition cycle [ILLERT, 1988]. This means that a pixel somewhere in the middle of the feature gets a high value, while a pixel on the border of the template gets a low value. He reported a success rate of 99% detecting a circular symbol.

FIGURE 51. EXAMPLE OF A PIXEL VALUE TEMPLATE FOR A CIRCLE



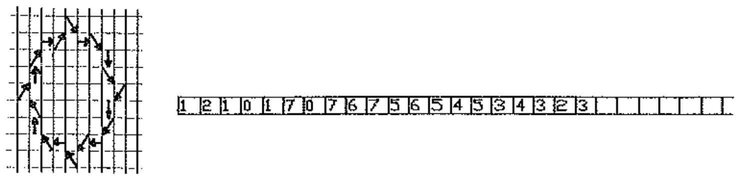
The method has several severe disadvantages. In the first place it is not rotation or scale invariant. It means that the symbols should be rotation invariant (circles, points) and that for every symbol size a different template should be used.

In the second place it is very computation intensive. This is increased by the requirement of using different templates for symbols which differ only in scale. It further poses restriction on the size of the symbol which should be detected because computation times increase quadratic with increases in template size.

C8.3.1.2 AUTOMATIC ATTRIBUTE CODING USING FREEMAN CODES

With Freeman codes a similar method can be applied. Only now the template does not consist of pixel values, but of Freeman codes. Because Freeman codes only apply on contours, the template has a linear form (see FIGURE 52). The process comes down to dragging the template along the contours of the image and computing the rate of similarity.

FIGURE 52. EXAMPLE OF A FREEMAN CODE TEMPLATE OF A CIRCLE

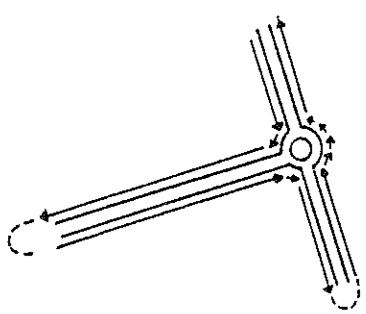


C8.3.2 AUTOMATIC ATTRIBUTE CODING USING CONTOUR VECTORS

A slightly different way of doing this is to represent the contour by vectors instead of Freeman codes. It is possible then to construct a function which gives the amount of angular bend between the starting point and a certain point on the contour. When this function is normalized (the starting point gets an angle which equals 0) and its domain is divided into equidistant steps then the matching process has become rotation and scale invariant [ILLERT, 1988].

Illert reports a success rate of 95%. It must be stressed however that drawbacks still exist. Although computation times will be considerably less then in the first method, it is our conviction that they still pose a restriction on the size of the object to be detected. A more serious drawback is that the object should be isolated. When it is intersected by another feature it is impossible to find the point where the contour of one object ends and the contour of the other one begins. Then the two objects will appear to the computer as one, higher order object. This means that the contour parts together making up the object to be detected will be spaced far apart in the contour describing the higher order object (see FIGURE 53).

FIGURE 53. SITUATION WHEN THE OBJECT TO BE DETECTED IS INTERSECTED BY ANOTHER OBJECT



C8.3.3 AUTOMATIC ATTRIBUTE CODING USING NUMERICAL PATTERN RECOGNITION

Numerical pattern recognition is a method widely used in the field of remote sensing. Here every object is determined according to certain properties. The object then is mapped into an n - dimensional feature space, in which n equals the amount of properties used. Techniques like box-, minimum distance - and maximum likelihood classification can be applied to detect different classes of objects. To determine the properties of an object Illert uses the so called grid method, in which a grid is laid over the object. Properties which can be derived from this are number of intersection with the cell sides, total arc length within a grid cell, amount of background in each cell etc. The feature space will then have the used properties together with the row and the column number of the cell as dimensions. Illert used only the arc length as a property and he reports a success rate of 95 %. He states however that there were some problems with normalization of the orientation. Which these were are not explained.

C8.4 COMPLEX OBJECT CODING

One of the highest-level tasks to carry out is to create objects with attributes which represent the data extracted from the map. Up to this stage, we have been able to extract the topology of the map, and cartographic objects. But there is more information to extract and more codes to create for the final GIS. At this stage unable to make very basic distinctions such as between a forest and a lake, or a road and a canal. We have extracted topological and geometric information (lines and regions), but we still have to express simple statements such as 'it is a forest' and 'it is a hydrographic object'.

In some cartographic organizations, maps are produced using (colour) separates, and GISs or GDBs (Graphic Data Bases) are structured into levels. In such an organization the basic operation in Automatic Digitizing project is to create 1 database level for each map separate. Colour separates thus evolve naturally into database levels during digitizing, and it is only necessary to extract information separately for each level. This means separate scanning and separate processing. At the end It is easy to attach to the Blue separate an attribute 'Water' or 'Hydrography', and to the Green separate 'Vegetation'.

Automatic valuation of digital contour lines is very common practice for the creation of Digital Elevation Models. By the use of rules, it is possible to label each contour line with its height [MOR, et al., 1972]. The (obvious) rule is that if a contour line has elevation Z, all its neighbouring contour lines have elevation Z - Dz, Z, or Z + Dz where Dz is the height interval between 2 contour lines on the original map.

More involved attribute coding problems exist, such as what is the difference between a motorway and a path, or what is the difference between a canal and a river, or what are the names of this valley and of the path along its bottom?

C8.5 POINT SYMBOLS AND TEXT

Point symbols and text differ in that sense from all the other symbols, that they are frequently composed of segments, which contain short vectors and have unequal line thickness.

The length of the vectors in a segment can be used to detect these symbols. To make distinction from deviating line symbols we must use the non repetitiveness of these symbols.

A lot of research has been done in the field of automatic recognition of text (Optical Character Recognition OCR)) (See for references: KAHAN et al., 1986). One can say that the concept of thinning has been developed for the sake of OCR. It is our belief that the results of these studies not only can be applied on text but also on point symbols. On a map the basic properties of text and symbols are the same. They are quite small and, compared to the other symbols on a map can have a rather complex topology.

The most obvious method to recognize text is template matching. However, we will not discuss this any further as the disadvantages have been outlined above.

A second method applies the topologic and geometric description of the symbol. After thinning the image, the skeleton is turned into a graph, which is made up of vectors and their topology. The symbols then are examined according to properties like number of nodes, number of arcs and number of holes. However, these properties are not sufficient to distinguish between every possible symbol. This is why higher level properties are being taken into account such as corners (right angles), and curves. But detection of curves, especially, still poses a problem. For example, recognition of arabic writings is still very problematic.

When this method is applied on map symbols we can make use of numerical pattern recognition methods. We then can map the symbols in a feature space whose dimensions are the above mentioned properties

A third approach takes into account the contour of the character. This approach has also been dealt with above. We cannot use it for maps because characters often are connected with other features.

One of the properties of text is that it appears in groups. The first criterion to detect these groups should be the distance between the characters of the group. Also a direction orientation should be applied. For both of these measures we need a character position. The center of gravity of the characters or the center of gravity of their maximum bounding box yield these positions. With capitals the latter one will give best results, while for lower cast letters the former one will do best. Other criteria can include properties like size and possible combinations of characters. These techniques are succesfully applied in the system used by the City of Vienna [WILMERSDORF, 1989].

C9. OTHER APPROACHES

The whole process of information extraction from digitized images is dependent on current brainware, software and hardware available. But the question arises: 'Are we doing the right thing ?' What should we do if our whole approach takes us down a sidetrack - a sidetrack in the network of informatics and human knowledge? Skeletonization, as we know it and as we have described it, is far from an ideal solution. Can we think of other solutions? Of course we can and we must, and that is what workers have been doing, since the early 1980s.

C9.1 SCANNING REVISITED

Returning to our origins - scanning. Scanning is basically a sampling process. From a continuous signal in the real world, only a few finite and discrete samples are selected. Deutsch [DEUTSCH, 1972] has proposed hexagonal tesselation instead of the common square pixels. Such a choice has two advantages:

> More isotropy. An element on the Hexagonal Grid (HG) has 6 neighbours. All neighbours have an equal status (while a square pixel has 4 direct neighbours and 4 diagonal neighbours);
>
> A more compact a structure. The HG wastes about half as much 'empty' space in the plane as the square grid, if we assume that both the square pixel and the hexagonal pixel provide information about a circular area fitting INSIDE the pixel [CARDOSO, 1988].

C9.2 HEXAGONAL GRID PROCESSING

The operations performed on the square grid can be designed for hexagonal grids. Erosion and dilatation, contour extraction and even skeletonization and line extraction can be easily developed. Sobel contour following [SOBEL, 1978] and Pavlidis [PAVLIDIS, 1980] contour thinning can be directly adapted to the 6-neighbourhood.

C9.3 HOUGH TRANSFORM

Hough transform was first described by Hough in 1962 [HOUGH, 1962]. It came into wider use after the publication of a paper by Duda and Hart [DUDA et al., 1972]. The basic idea of Hough transform is that a line is made up of points which are related to each other by a directional link.

A straight line can be characterized by 2 parameters :

> an angle Theta;

and,

> a distance r

The Hough transform is expressed by the theorem:

> two straight line segments belong to the same straight line if, and only if, they have the same parameters Theta and r.

We will not develop this basic idea. Detailed explanations of Hough transforms can be found in several papers [BALLARD, 1981; STOCKMAN et al., 1977; DAVIES, 1987a; DAVIES, 1987b; SKINGLEY et al., 1987; ROSENFELD, 1983].

Many researchers have conducted work on Hough transform and developed it. It is a parallel algorithm, and implementations using dedicated parallel hardware have been considered. Applications to remote sensing have been tested (e.g. in SAR imagery). But many problems remain and too many problems are not addressed, which hinder the use of Hough transforms in map digitizing; an example using Hough transforms is given below:

FIGURE 57. HOUGH TRANSFORM OF A TEXT CHARACTER

the detected lines
are not necessarily
useful.

C9.4 HIERARCHICAL IMAGE ANALYSIS

In an image, objects can be roughly described as being large or small. It is possible to say that a given object is larger than another one. Small objects are 'details'. Hierarchical image analysis proceeds by first detecting large features in the image and then refining them by detecting smaller details. The aim is to imitate human (biological) vision and image understanding. Tools for hierarchical analysis are still at the research stage. The names given to these tools are 'image pyramids', 'magnifying-glasses', [ROSENFELD, 1983].

C9.5 HUMAN VISION

We know there is at least one successful solution: the biological one. Endeavours must be made to study and copy this solution - and perhaps this will lead to a suitable approach in digital data acquisition.

C10. PHASES II AND III CONCLUSIONS

Our study covering Phases II and III has been based on the steps shown in FIGURE 17 of section C1. However the questionnaires summarised in part A of this report raised the problems of vectorizing and related batch processing (loss of detail, distortion of detail, introduction of artefacts), and automatic feature coding (recognition of lines, recognition of points). In this concluding section of Part C we must summarise our findings within the structure of FIGURE 17, and then relate these findings to the problems raised by the questionnaires (these problems are itemised in section C1 as 1.1 - 5.2) in part A.

C10.1 FINDINGS WITH REGARD TO FIGURE 17

We found 'noise removal' approaches assumed that noise had small area (C3.1.2), small perimeter (C3.1.3), or a shape consisting of a chain of short vectors beginning and ending near each other (C3.2.2). This last approach we successfully implemented (C3.2.1.1 and C3.2.1.2) and it reduced image noise (both 'salt and pepper' noise and that intersecting valid features) considerably.

Another form of noise, introduced by the automatic digitizing process rather than the original image, is bridging. We proposed a three step approach to this: 1) automatically identify where bridging is likely to happen; 2) confirm the presence of bridging by detecting 'white noise'; and, 3) convert bridge pixels to background pixels enabling subsequent thinning of the bridged objects as two separate objects (C3.3).

Considering 'skeletonization', we found that weaknesses in the most frequently implemented 3x3 matrix analyses of the scanned data (to identify background, edge, and interior pixels) resulted in the creation of disconnected objects (C4.1). Various solutions, including: directional subpasses (which introduced small directional errors); using larger matrices (which resulted in slow processing times); the Xia algorithm; and the Pavlidis algorithm were examined. We invested considerable effort in implementing and testing the Pavlidis algorithm, and found it an effective algorithm; especially noteworthy was its ability to handle lines of uneven thickness, to preserve the shapes of small corners, and to avoid the introduction of directional bias. However under some circumstances problems remained, as shown in the FIGURES 18-25. Modifications were introduced to our implementation of the Pavlidis algorithm, which we believe are effective (C4.2.2.2).

After 'skeletonization' on FIGURE 17 can be seen 'node improvement'. The apparently intractable problem of junction distortion (node displacement) seems only to arise for those in the mapping industry. Other users of scanned data remain unconcerned - so we in the mapping business have to find our own solutions. However we did not go straight to node improvement, but instead developed a technique for 'skeletonization with node improvement' A brief review of our work on this technique follows.

Junction distortion arises because the medial axis concept collapses at junctions (C5.2), so we must find an alternative. First to identify junctions; three approaches are described: Domogalla's (C5.1) and two of our own which use Freeman codes and vectors (C5.5) respectively, and also clean noisy junctions. Once the junctions have been found (and cleaned if necessary) the true node positions can be determined. A detailed analysis of the causes of node displacement and the means for its correction are given in sections C5.3 and C5.4. All our findings relating to skeletonization with node improvement were successfully implemented in the TON (thinning of nodes) algorithm (C5.6).

We next addressed 'line tracking' in C6. Once a proper skeleton has been formed (the subject of C4 and C5) then line tracking is relatively easy, although present methods seem time consuming. We developed a fast and error free approach (C6.1, C6.2) for line tracking between nodes, and also proposed a procedure for organising the whole line tracking (vectorization) task for an image (C6.2.1).

'Segment merging' is required when patching has been introduced to ensure that the processor does not have too much data to handle at any one time, or when data is misleadingly quantised as a result (for example) of pecked lines on a map representing a continuous feature - such as a footpath.

The merging of patches can be dealt with in both raster and vector mode (C7.1.1 and C7.1.2), and although we proposed approaches to the problem they were not implemented or tested - partly because our experiences with SysScan software lead us to believe it is not a problem.

More important is the ability to merge lines broken by their map symbolisation. We found processing while the data was still in raster form to be unsuccessful (C7.2.1), but an analysis of the problem caused by the rather simple line styles on maps and while the data was in vector form indicated an algorithm which we successfully implemented (C7.2.2 and C7.2.3).

Our approach to 'topological reconstruction' was mainly a literature review. Topological reconstruction requires firstly that arcs are constructed between nodes - and the line tracking solutions addressed this. We looked (C8.1.1 and C8.1.2) at both vector and raster segmentation procedures - especially those which linked arcs of similar colour (as detected by the scanner) or line-weight (as detected by the thinning process).

An aspect of Topological reconstruction is 'automatic attribute coding'. Automatic attribute coding can be aided by component labelling (C8.2.3), but look-up-tables linking the automatically generated component labels to their true labels have still to be created interactively. An often quoted approach to automatic attribute coding is template matching. The template may describe the object to be matched in Freeman codes or vectors (C8.3.1.1, C8.3.1.2, C8.3.2), but disappointingly we found problems associated with size and rotation variability, and also confusion between objects. De Simone's top-down approach which he presented in 1986 does not appear (in the literature) to have undergone further development, and we are still left with the solutions available ten years ago - successful contour coding but otherwise a need to scan separates. However certain users (e.g. Magistratsdirektion der Stadt Wein) having clean large scale map documents, are satisfied with the de Simone approach, as implemented by SysScan.

C10.2 FINDINGS WITH REGARD TO POINTS RAISED BY THE QUESTIONNAIRES

We must now consider the problems posed by the questionnaires, and see how many of them have been answered!

1. Loss of detail

1.1 How can we decide the resolution at which a map should be scanned, taking into account e.g. linewidth, gap width, scanning times, interactive editing times, etc.?

In the discussion on bridging in C3.3 we proposed a modification of the 'rule of thumb' presented in B2.1 to read "scanner resolution should be at least half the minimum detail width appearing in the document". The reader will realise that this may increase scanning and processing too much - especially if only a few examples of fine detail exist. The operating staff will have to decide (based on their experience, the number of locations where problems will arise, and the length of time typically taken to correct such faults) whether interactively correcting faults caused by low resoution is more efficient than scanning at a higher resolution.

1.2 How can automatic cleaning of the raster data set be performed without loss of useful information?

As described in the second paragraph of this section we have outlined techniques which identify both 'salt and pepper' noise and noise on valid features. These work successfully, but include constants which have to be correctly estimated and set by the user. They assume that the noise is smaller than a valid feature, otherwise useful information is lost. These approaches could be used to remove screen dots from a scanned map.

1.3 How do we cope with contour clustering, assuming an otherwise optimal resolution is used?

See the second paragraph of this section.

2. Distortion of Detail

2.1 How serious in metric terms, is intersection node displacement when a thick and thin line intersect? How does this arise? If this is a serious problem, how can it be overcome?

We found that this was likely to be a serious problem once one line was more than three times as thick as the other, but modifications of the TON and associated algorithms could increase this ratio to 6:1, or better. The details of this are in section C5.3.

2.2 How serious in metric terms, is intersection node displacement when one or both of the intersecting lines is curved? How does this arise? If this is a serious problem, how can it be overcome?

We found that intersecting curved lines could be treated as intersecting straight lines. The details of this are in section C5.2

2.3 How do we re-establish the correct corner angle when vectorizing has distorted it?

We have found this to be an effect of skeletonization, but careful use of the Pavlidis thinning algorithm will prevent the problem from arising. The details of this are in section C4.2.2.2.

2.4 How can it be ensured that the correct dangling node is found?

We did not address this problem specifically, but suggest that when an isolated end point is found (i.e not resulting from patching or line symbolisation on the original map) then its true end point will be beyond the existing end point in the direction of the line, and a distance from that end point resolved to n x pixels, where n is the number of peeling operations which have been performed on the line during skeletonization.

3. Introduction of Artefacts

3.1 How can it be ensured that bridges do not form or are successfully removed at angled intersections?

See 1.3 above.

3.2 How can it be ensured that bridges do not form or are successfully removed at the locations containing fine detail, such as close parallel lines, text, point symbols and point symbols incorporated into patterned lines?

As already indicated (3.1 above) it is possible to predict where bridging might occur and search the data set for those occurrences. Text, when isolated from other features can be recognised and removed from the dataset using OCR (Optical Character Recognition) techniques. These techniques are less successful with point symbols. Techniques for handling point symbols forming parts of patterned lines are described in section C7.2.2.2, but in this case the actual point symbols do not have to be recognised because it is the line which has to be recognised.

3.3 How can artefacts be prevented from being generated when lines of uneven thickness are vectorised?

We have found this to be an effect of skeletonization, but careful use of the Pavlidis thinning algorithm will prevent the problem from arising. The details of this are in section C4.2.2.

3.4 How can it be ensured that bridges do not form when a line is highly tortuous?

We did not specifically address this problem, but it can be suggested that the problem arises when the gap between two close parts of the highly tortuous contour is smaller than twice the scanning resolution. The 'rule of thumb' mentioned in the third paragraph of this section (i.e. under 1.1) is useful.

4. Recognition of lines.

4.1 How can we ensure that any line made from pecks, point symbols etc. can always be recognised as a line in spite of gap sizes elsewhere on the map and the tortuosity of the line itself?

We have assumed that the line patterns on maps are rather simple, and have successfully implemented an algorithm to recognise the line pattern. As yet the algorithm does not handle the repeating nature of the pattern. This algorithm is described in sections C7.2.3.

4.2 How can we ensure that lines crossing a patterned background (screens, hatching, hillshading) are recognised?

In the case of screens, the dots can be removed as noise (see 1.2 above), otherwise we have not solved the problem.

4.3 How can we ensure that line continuation is found despite gaps such as those formed by deleted contour values or the partial elimination of contours in areas of steep terrain?

We have not addressed this problem. Lack of time gave it a low priority. This low priority is justified because scanned contours are nowadays frequently used to generate DTM's, and gaps in the original contours are unimportant, and the techniques which automatically code contours can cope with gaps. Once contours are either correctly coded, or part of a DTM, gaps can be filled.

4.4 How can we ensure that feature codes based on linestyle, weight, colour, pattern, or context can be attached to lines?

Colour may be recognised at the time of scanning. Weight can be determined at the time of thinning. Certain simple linestyles can be recognised (C7.2.2). Otherwise we have not addressed this problem.

5. Recognition of Point Symbols

5.1 How can we ensure that simple geometric features such as circles, arcs, crosses, text, utilities' symbols etc,. are correctly recognised and identified, even when they are close to line symbols or each other, oriented at different angles, and of varying size?

Although we addressed this problem in sections C8.3.1.1, C8.3.1.2, and C8.3.2, we are not convinced solutions exist.

5.2 Which pattern recognition tasks are best performed when the data are still in raster form and which benefit from the data being in vector form?

Our answer must relate to our findings in 5.1, above. It appears that the vector approach is more successful (C8.3.2), reports of 95% success are recorded, but only on isolated objects.

C11. THE FUTURE

We have answered almost all the questions raised by the questionnaires. Those still remaining could be addressed as student projects, and reported on in the literature. It is also hoped that some of the algorithms we have developed will be more completely examined by others. Although we have answered most questions, the answers may not be satisfactory. Automatic feature coding is especially disappointing. According to the literature automatic feature coding is not very effective - however there may be some reluctance to publicise succes in this area.

One thing that has emerged is that there are many different approaches to solving the same problem, and during our visit to the Institute for Digital Image Processing and Computer Graphics (DIBAG) in Graz, this was confirmed. There, now, the approach is very much to find the appropriate solution to each digitizing situation or problem. This is also confirmed in the VTRAK (Laserscan) approach, where the user is supplied with a large collection of automatic, semi-automatic, and interactive tools to solve digitizing problems.

Since this Working Group became operational in the mid 1980's three trends have emerged. The first (low cost scanners) we addressed in PHASE I. The second (operators must flexibly apply a range of tools to the data capture problem) is supported by almost all our findings in PHASES II and III, and the work of others mentioned in the preceding paragraph. The third is the emergence of 'on screen digitizing'. One might regard it as a desperate reponse to total automation's failure. Alternatively one might regard an analysis of 'on screen digitizing' as PHASE IV of our work; but we will not. It can be PHASE I for another Working Group.

LITERATURE

Arcelli, C., 1981, Pattern thinning by contour tracing, In: Computer Graphics and Image Processing 17, pp. 130-140.

Arcelli, C., Sanniti di Baja, G., 1984, An approach to figure decomposition using width information. In: Computer Graphics and Image Processing 26, pp. 61-72.

Ballard, 1981, Generalizing the Hough transform to detect arbitrary shapes, In: Pattern Recognition, Vol. 13, Nr. 2, pp. 111 -122.

Beck,F.J., and R.W.Olson "Quality Control and Standards for a National Digital Cartographic Database" Proceedings AutoCarto London, Royal Institute of Chartered Surveyors, 1986.

Blum H., 1964, A transformation for extracting new descriptors of shape, Symposium on models for the perception of speech and visual form, MIT Press, Cambridge, Massachusetts.

Bouille, F., 1980, General methodology of map digitization and some improvements based upon pattern Recognition. ICA, Tokyo, 1980

Bouille, F., 1988 Personal communication, ENSG, IGN.

Bouille F., 1988, Developing strategies in GIS by problem-solving methods based on a structured expert system. In: Muller, J.C. (ed.), Eurocarto 7, Envoronmental applications of digital mapping, ITC publication Nr. 8, ITC, Enscede, Holland.

Byte, (Journal) July 1988, p.181-185

Cardoso, 1988, Signal theory, ENSG lecture notes.

Cartography, 1988, Journal of the Australian Institute of Cartographers, Vol. 17, Nr.2.

Chung, Y.S. "Evaluation of the DTM data acquisition using low cost scanner". ITC M.Sc. Report, Enschede, 1989.

Clark, R. "Cartographic raster processing programs at USAETL." Proceedings of the 40th annual meeting of the ACSM, St. Louis, March 1980.

Davies E. R. and Plummer A. P. N., 1981, Thinning algorithms : a critique and a new methodology, In: Pattern recognition, Vol.14, pp. 53 - 63.

De Simone M., 1986, Automatic structuring and feature recognition for large scale digital mapping. In: AUTO CARTO 6, London, Great Britain.

Denegre, J., Foin, P., 1978, Acquisition, processing and management of digitized cartographic data: the IGN - France computer system. In: Van Zuylen L. (ed.), Computer assisted cartography, Nairobi, Kenya, pp. 166 - 178.

Desk Top Publishing, Jan. 1988

Deutsch E.S., 1972, Thinning algorithms on rectangular, hexagonal and triangular arrays.In: Communications of the ACM, Vol.15, Nr. 9, pp. 827 - 837.

Domogalla U., 1984, Entwicklung eines kompakten Systems zur Digitalisierung komplexer graphischer Vorlager, DIBAG-Bericht N.20-1 (in German) from the Institute for Image Processing and Computer Graphics, A-8010 GRAZ, Austria

Drummond, J., Raster processes in automated map production, Enschede, ITC: Dept. of Cartography, 1986.

DrummondJ.E.,H.M.Bosma,L.Raidt:"APreliminaryInvestigationintoaLow-Costscanner" ITC Journal 1989 nr 2

Duda R. O., Hart P. E., 1972, Use of the Hough transformation to detect lines and curves in pictures, In: Communications of the ACM, Vol. 15, Nr.1.

Eidenbenz, C., 1989, Scannertechnik zur Erfassung van Planen und Karten. In: Vermessungm, Photogrammetrie, Kulturtechnik 2/89.

Espelid R., 1988, A raster-to-vector conversion concept based on industrial requirements, IAPR Workshop on CV, Special hardware and Industrial applications, Tokyo, Japan.

Freeman, H., 1961, One the encoding od arbitrary geometric configuration. In: IRE Trans EC, june, pp. 260 - 268.

Haldrup, K. "Scanning af kort og ledningsplaner - Naturgas Syd" 1987 Dagmarhus, Geoplan, Copenhagen

Hough P. V. C., Method and means for recognizing complex patterns. U.S. patent 3 069 654, 18 December 1962

Houston Instrument, Hi-Scan Manual. Appendix C-3, 1987

Houston Instrument, ScanCAD Attachment Manual. 1988

Illert, A., 1988, Automatic recognition of text and symbols in scanned maps. In: Muller, J.C. (ed.), Eurocarto 7, Environmental applications of digital mapping, ITC publication nr. 8, ITC, Enschede, Holland.

Kasvand, 1986, Linear textures in line drawings, In: IEEE (?).

Kahan, S., Pavlidis, T., and Baird, H.S., 1987 On the recognition of printed characters of any font and size. IEEE Transactions on Pattern analysis and Machine Intelligence4. Vol PAMI 9 No2. March 1987.

Kers A.J., Cartography Department ITC, personal communication.

in Knipselkrant: Feb 1988

Kwok P.C.K., November 1988, A thinning algorithm by contour generation. In: Communications of the ACM, Vol. 31, Nr. 11

Landy, M.S., Cohen, Y., 1985, Vector graph coding: Efficient coding of line drawings. In: Computer Graphics and Image processing 30, pp. 331 - 344.

Lauenstein,W.,K.Meinecke,G.Otschik:"Schwarz-Weiss-ScannerGerateundEinsatzgebiete - eine Checkliste" Technik+Forschung,Weisbaden 1987 - IV

Lummaux, oral communication, 1988, IGN

MacUser, (Macintosh Users Journal) Jan. 1989

MacUser, (Macintosh Users Journal) Sep. 1988

MacUser, (Macintosh Users Journal) Aug. 1988

Martinez-Perez Pilar, M., Jimenez J., Navalon J.L., 1987, A thinning algorithm based on contours. In: Computer vision, graphics and image processing 39, pp. 186 - 201.

Meadow,A.R.,R.Offner,M.Budiansky:"HandlingImageFileswithTIFF" DrDobbsJournal of Software Tools: May 1988

Mor M., Lamdan, T., 1972, A new approach to automatic scanning of maps. In: Communications of the ACM, Vol.15, Nr.9.

Naccache, N.J., Shinghal, R., 1984, SPTA:AProposed Algorithm for Thinning Binary Patterns. In: IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-14, No3, 1984

Nazif, A.F., Levine, M.D., 1984, Low level image separation, an expert system. In: Pattern analysis and machine intelligence 6, Nr. 5, pp. 555 - 557.

Nagao, T. Agui, T., Nakajima, M., 1988, An automatic road vector extraction method from maps. In: 9th ICPR, IEEE, Rome, Italy.

Parker, J.R., 1988, Extracting vectors from raster images. In: Computer & Graphics, Vol 12, Nr. 1, pp. 75 - 79.

Pavlidis T., 1980, A thinning algorithm for discrete binary images. In: Computer graphics and image processing 13, pp. 142 - 157.

Pavlidis T., 1980, A thinning algorithm for discrete binary images. In: Computer graphics and image processing, V.13, pp. 142 - 157.

Pavlidis T., 1981, A flexible parallel thinning algorithm. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol PAMI - 3 , pp. 162 - 167.

Pavlidis, T., 1982, An asynchronous thinning algorithm. In Computer graphics and image processing 20, pp. 133 - 157.

Pavlidis, T., 1982, Algorithms for graphics and image processing, Washington DC, Computer Science Press.

Pavlidis, T., Van Wijck, C., 1985, An automatic beautifier for drawings and illustrations. In: SIGGRAPH Conference proceddings, San Fransisco, USA.

Peucker T. K. and Chrisman N., 1975, Cartographic data structures, In: The American cartographer, Vol. 2, Nr. 1, pp. 55 - 69.

Peuquet D. J., 1981, An examination of techniques for reformatting digital cartographic data. Part 1 : the raster-to-vector process. In: Cartographica, V. 18, Nr. 1, pp. 34 - 48.

Pickering, R.P., 1988, The process of vectorisation in computer assisted map production. In: Personel study topic reports 1988, ITC, Enschede, Holland, pp. 81 -95.

Ramer, U., 1972, An iterative procedure for polygonal approximation of plane curves. In: Computer Graphics and Image processing 1, pp. 244-256.

Rosenfeld A., 1970, Connectivity in digital pictures. In: Journal of the Association for computing machinery, Vol.17, Nr.1, pages 146 - 160.

Rosenfeld A., 1975, A characterization of parallel thinning algorithms. In: Information control, Vol. 29 Nr. 3, pp. 286 - 291.

Rosenfeld A., Pfaltz J. L., 1966, Sequential operations in digital picture processing. In: Journal of the Association for computing machinery, Vol. 17, Nr. 1, pp. 146 - 160.

Salge F., Sclafer M. N., 1988, The IGN cartographic database from the users' needs to the relational structure. In: Muller, J.C. (ed.), Eurocarto 7, Environmental applications of digital mapping, ITC publication Nr. 8, ITC, Enschede, Holland, pp. 51 - 65.

Salge F., Piquet-Pellorce D., 1986, The IGN small scale geographic database. In: Autocarto 6, London, pp. 51 - 65.

Siemens, CADScan product information, CADSCAN CON Smart Thinning Algorithms.

Stefanelli R. and Rosenfeld A., 1971, Some paralllel thinning algorithms for digital pictures. In: Journal of the Association for computing machinery, Vol. 18, Nr. 2, pp. 225 - 264.

Satoshi S., Keiichi, A., 1986, Sequential thinning of binary pictures using distance transformation. In: IEEE, pp. 289 - 292.

Scan-Cad Attachment Manual, 1987 (Houston Instrument)

Slansky, J., 1978, On the Hough technique for curve detection. In: IEEE Transactions on Computers, Vol. c-27, nr. 10.

Smith R., 1987, Computer processing of line images : a survey. In: Pattern recognition, Vol. 20, Nr. 1, pp. 7 - 15

Sobel I., 1978, Neighbourhood coding of binary images for fast contour following and general binary array processing, In: Computer graphics and image processing 8, pp. 127 - 135.

Spani B., 1990, Scannen von planen und dann? Eine Zwischenbilanz, In: Vermessung, Photogrammetrie, Kulturtechnik 5/90.

Stockman G.C., Agrawala A.K., 1977, Equivalence of Hough curve detection to template matching. In: Communications of the ACM, Vol.20, Nr.11

Toumazet J. J., Traitement numerique de l'image (in French).

Tamura H., 1978, A comparison of line thinning algorithms from digital geometry viewpoint. In: 4th ICPR, Proceeedings, Kyoto, Japan, pp. 715 - 719.

Thompson,C.N., Test of Digitizing Methods. OEEPE Publication No.14, Institut fur Angewandte Geodasie, Frankfurt, 1984.

Waters, R., Meader, Dr. R., Reinecke, G., 1989, Data capture for the nineties : VTRAK. In: AUTO CARTO 9, Proceedings, Baltimore.

Wilmersdorf, E., 1989. A computer-assisted assembly line for the production of large scale maps. International Cartographic Association 14th. World Conference, Budapest, 1989.

Xia Y., 1986, A new thinning algorithm for binary images. In: IEEE Transactions in Pattern Analysis and Machine Intelligence, VOL PAMI-8 pages 995 - 997.

Yamada, H., 1984, Contour DP matching method and its application to handprinted Chinese character recognition. In: 7th ICPR, Proceedings Vol. 1, IEEE, Montreal, Cananda.

You, K.C., Fu, K.S., 1979, A syntactic approach to shape recognition using attributed grammars. In IEEE transactions SMC-9, Nr. 6, pp. 334 - 345. Arcelli, C., 1981, Pattern thinning by contour tracing, In: Computer graphics and image processing 17, pp. 130 - 140.

Appendix A

THE FIRST QUESTIONNAIRE

PRELIMINARY QUESTIONNAIRE

1. Organisation :
   Contact Person :
   Address :

2. Organisation's Products:

3. Do you use computer assisted techniques in your map production? YES NO

4. If the answer to (3) was 'YES', at what stage?

a) Document Management/Map Cataloguing	YES	NO
b) Map Design	YES	NO
c) Map Data Capture	YES	NO
d) Map Data Processing	YES	NO
e) Map Presentation	YES	NO
f) Map Archiving	YES	NO
g) Spatial Information System Management	YES	NO

(add more information if you wish)

5. If the answer to (3) was 'YES', do you use:

a) manual digitizing;	YES	NO
b) semi-automatic digitizing;	YES	NO
c) automatic digitizing;	YES	NO
d) purchased automatic digitized raster data;	YES	NO
e) purchased vector data, without feature codes, or,	YES	NO
f) purchased vector data, with feature codes.	YES	NO

(add more information if you wish)

6. If you use automatic digitized raster data, do you vectorize the raster data?

YES NO

(add more information if you wish)

7. If you use vectorized or vector data do you feature code the data?

YES NO

(add more information if you wish)

8. If the answer to (7) was YES how many feature codes do you use per item?

Give number here:............

(add more information if you wish)

9. Do you add further intelligence (i.e. more than feature codes) to you spatial data, e.g. by using Database Management Software and connecting your spatial items to the database?

YES NO

(add more information if you wish)

10. Would you be willing to disclose any more about your digital mapping techniques?

NO MORE
A LITTLE MORE
SOME MORE
MUCH MORE

11. Thank you.

Automatic Digitizing Working Group
OEEPE Commission D

November, 1987

Appendix B

THE SECOND QUESTIONNAIRE

OEEPE COMMISSION D - AUTOMATIC DIGITIZING WORKING GROUP

SECOND QUESTIONNAIRE

PART 1 . IDENTIFICATION OF YOUR ORGANISATION'S PRODUCTION AREAS, AND ASCERTAINING WHETHER SCANNING IS USED.

1.1 It is assumed that scanning is part of your production process. Is this true?

YES NO

Or is your scanner still used only for development, testing, or research?

YES NO

Add more information, please.

1.2 What are the hardcopy* products of your organisation?

- A topographic maps;
- B thematic maps;
- C cadastral maps;
- D utilities maps;
- E atlases;
- F other (please describe).

Are any of these in digital form?

A B C D E F

In the production of which of these does scanning play a role?

A B C D E F

If possible, please give samples of your products produced using scanning.

1.3 What are the digital products of your organisation?

- A digital maps in raster form, with feature codes*;
- B digital maps in raster form, without feature codes;
- C digital maps in vector form, with feature codes;
- D digital maps in vector form, without feature codes;
- E digital maps in vector form linked to a data base;
- F other - please describe.

In the production of which of these does scanning play a role?

A B C D E F

hardcopy = output to film or paper; softcopy = output to a screen

---

PART 2 IDENTIFICATION OF THE PRODUCTION STAGES IN YOUR ORGANISATION WHERE SCANNING IS APPLIED.

2.1 Which types of document do you scan?

A. Original fieldwork documents on paper;

A1 - in 'lead' pencil
A2 - in coloured pencil
A3 - in ink
A4 - other (please describe)

B. Original fieldwork documents on polyester;

B1 - in 'lead' pencil
B2 - in coloured pencil
B3 - in ink
B4 - other (please describe)

C. Image/photo interpretation overlays on paper;

C1 - in 'lead' pencil
C2 - in coloured pencil
C3 - in ink
C4 - other (please describe)

D. Image/photo interpretation overlays on polyester;

D1 - in 'lead' pencil
D2 - in coloured pencil
D3 - in ink
D4 - other (please describe)

E. Existing monochrome paper maps;
F. Existing monochrome maps on polyester;
G. Existing colour paper maps;
H. Large Scale maps (>1:5000);
I. Medium Scale maps (1:5000 to 1:100000);
J. Small Scale maps (<1:100000);
K. Specially prepared documents for scanning, please describe;

L. Existing production films (or pulls);
M. Film separates used in making production films;
N. Other, please describe.

Please, if possible, provide samples of your input documents.

2.2 For the digital spatial* data you do not scan how do you acquire it?

A. Buy it;
B. manual line following digitizing;
C. field observations;
D. classified of raster remote sensing data;
E. photogrammetric digitizing;
F. donated;
G. free data exchange;
H. other (please describe).

2.3 What companies/company manufacture(s) your scanner(s)?

spatial data = data whose position in space has been referenced by an x,y (and perhaps z) coordinate system.

PART 3 . ASSESSMENT OF SCANNING

3.1 Has your organisation benefitted from the introduction of scanners into your work procedures?

VERY-MUCH SOMEWHAT NOT-AT-ALL

3.2 If you gave VERY MUCH or SOMEWHAT above, how did you benefit?

A.	FINANCIALLY	VERY-MUCH	SOMEWHAT	NOT-AT-ALL
B.	GREATER STAFF INTEREST	VERY-MUCH	SOMEWHAT	NOT-AT-ALL
C.	WIDER MARKET	VERY-MUCH	SOMEWHAT	NOT-AT-ALL
D.	INCREASED PRODUCTION	VERY-MUCH	SOMEWHAT	NOT-AT-ALL
E.	BETTER QUALITY PRODUCTS	VERY-MUCH	SOMEWHAT	NOT-AT-ALL
F.	FASTER PRODUCTION	VERY-MUCH	SOMEWHAT	NOT-AT-ALL
G.	OTHER (PLEASE DESCRIBE)	VERY-MUCH	SOMEWHAT	NOT-AT-ALL

Please give further details.

3.3 Have you developed special techniques beyond those suggested by the manufacturer of your scanner, for example production techniques, extra software, etc., which have enhanced your system?

YES NO

3.4 If the answer to 3.3 was 'YES' please give further details.

PART 4 . PROBLEMS

4.1 Are you using your scanner(s) more than you had hoped when you purchased it(them)?

YES NO

or,

Are you using your scanner(s) just as much as expected when you purchased it(them)?

YES NO

or,

Are you using your scanner(s) less than you had expected when you purchased it(them)?

YES NO

4.2 If you are not using your scanner(s) as much as you had hoped to is this for:

A. Financial reasons;
B. Technical reasons;
C. Personnel reasons;
D. Educational reasons;
E. Other reasons (please specify).

Give further details, please.

4.3 Do you expect to increase the use of your scanners?

YES NO

4.4 If you answered 'YES' to 4.3 do you predict any problems when you increase the use of your scanners?

YES NO

4.5 If you answered 'YES' to 4.4, are the problems being addressed by:

A. your organisation;
B. the scanner manufacturer;
C. a consultant;
D. other, please specify.

4.6 If the answer to 4.4 was 'YES', what are these problems? Please explain.

4.7 Have you had some problems for which you have found solutions (for example to do with staffing, production rates, vectorizing)?

YES NO

4.8 If the answer to 4.7 was 'YES', describe them and their solution, please.

PART 5 . PRODUCTION PROCESSES

With the aid of a FLOW DIAGRAM, BLOCK DIAGRAM, or other such device, please describe your production procedures which involve scanning. Nodes in your flow diagram might be:

SKELETONIZED RASTER FILE; DOCUMENT PREPARED FOR SCANNING; ARCS/NODES RECOGNIZED; RASTER FILE AFTER SCANNING; etc.,

and will probably go from (for example) CLEANED ROAD SEPARATE to ERROR FREE FEATURE CODED* VECTOR DATA SET.

At each step in your diagram can you describe the procedures followed to reach this step, including any quality control procedures used.

You may wish to produce several different 'product lines' for different products.

FEATURE CODED = having a descriptor code attached - perhaps a level or layer, or code describing the item as a Class A road, a Parish boundary, a sewerage inspection cover, etc.

PART 6 TRAINING

6.1 Could you please describe the staff training programs you have found it necessary to implement since you introduced scanners.

6.2 With the introduction of scanners have your staff recruitment policies changed? Please explain.

PART 7 . WHAT NEXT?

7.1 Thank you very much for your cooperation sofar.

7.2 May we contact you further if there are still some outstanding points we would like to clarify?

YES NO

7.3 Might it be possible for one or two members of our working group to visit you in the future, especially to discuss findings in our research project?

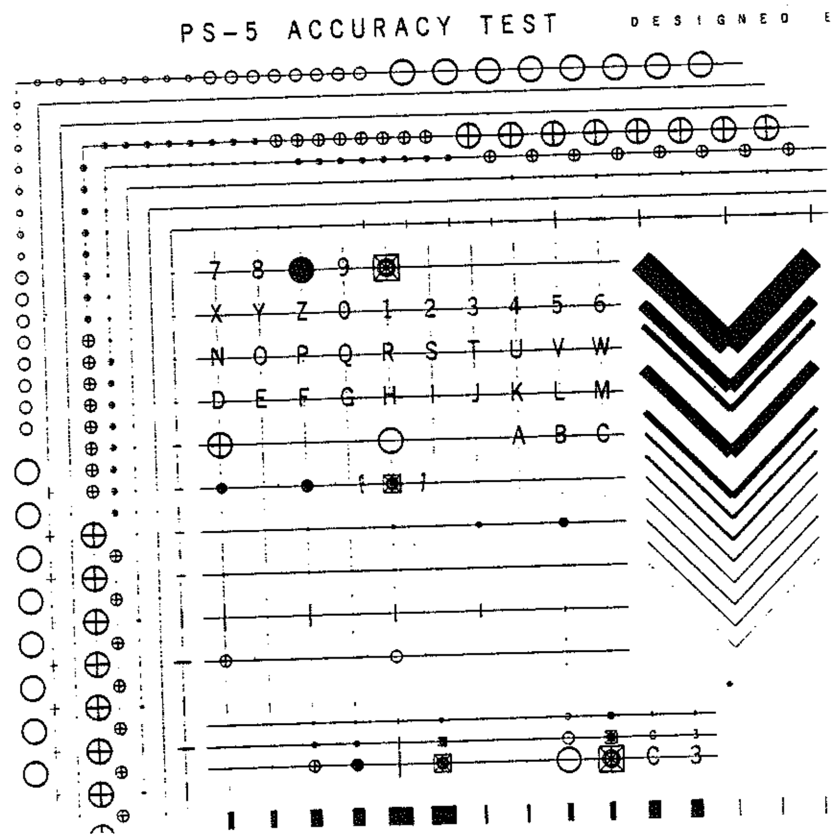
YES NO

THE THIRD QUESTIONNAIRE
(THE TEMPLATE)

EXAMPLES 1 & 2
Accuracy test (PS-5 plottertest for Sysscan)
line-art, black & white, positive, photographic emulsion
EXAMPLE 1: stable, clear film.
EXAMPLE 2: stable, opaque paper.

* How would this grid be distorted with your scanning process and do you correct this?
* Which is the thinnest line to show completely with your scanning process?

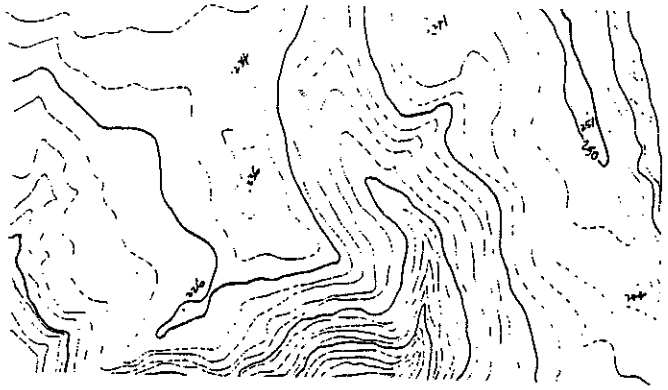
General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?

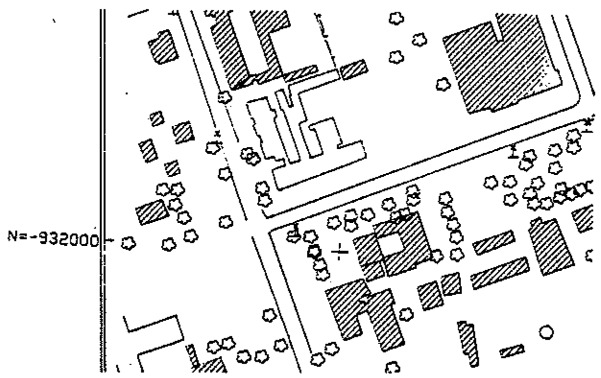


EXAMPLES 3 & 4
Photogrammetric plot (ITC exercise)
line-art, black & white, positive, pencil.
stable, translucent polyester.
EXAMPLE 3: roughly sketched contours.
EXAMPLE 4: crisp sketch of details.

* What line quality do you need?

General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?

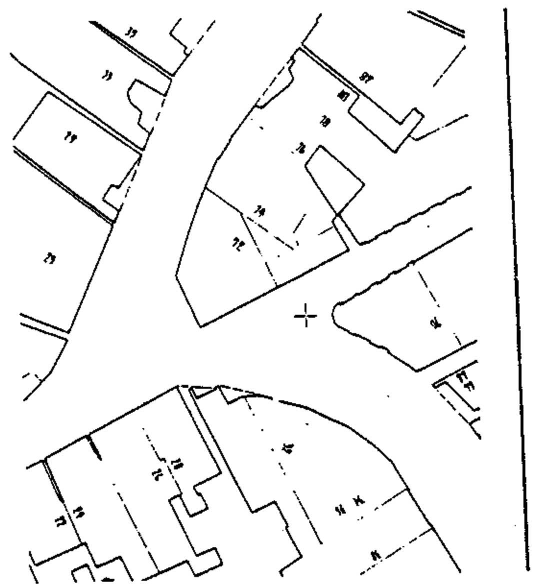




EXAMPLE 5
Large scale topographic map
("1:500" Openbare Nutsbedrijven Enschede)
line-art, black & white, positive, diazo-copy.
unstable, translucent.

* How do you reconstruct the lines of this poor original?

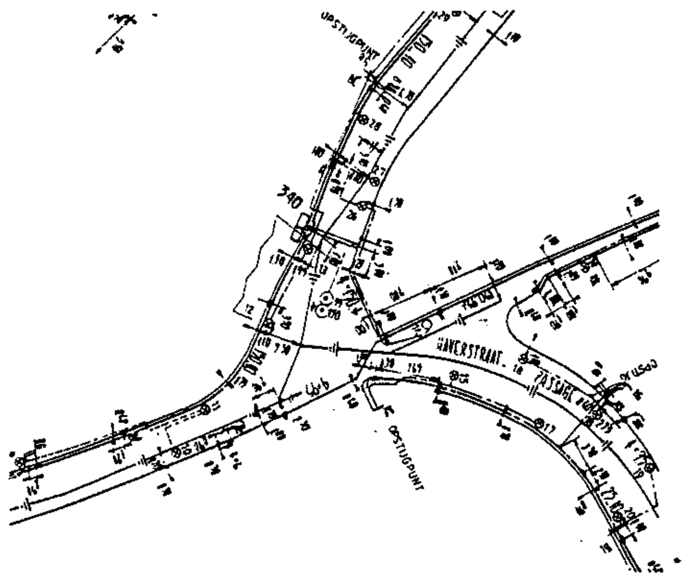
General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?



EXAMPLE 6
Utility ("Laagspanning" Openbare Nutsbedrijven Enschede)
line-art, black & white, positive, diazo-copy,
unstable, translucent.

* What problems do you have with parallel lines?
* What problems do you have with stencil or hand drawn text?

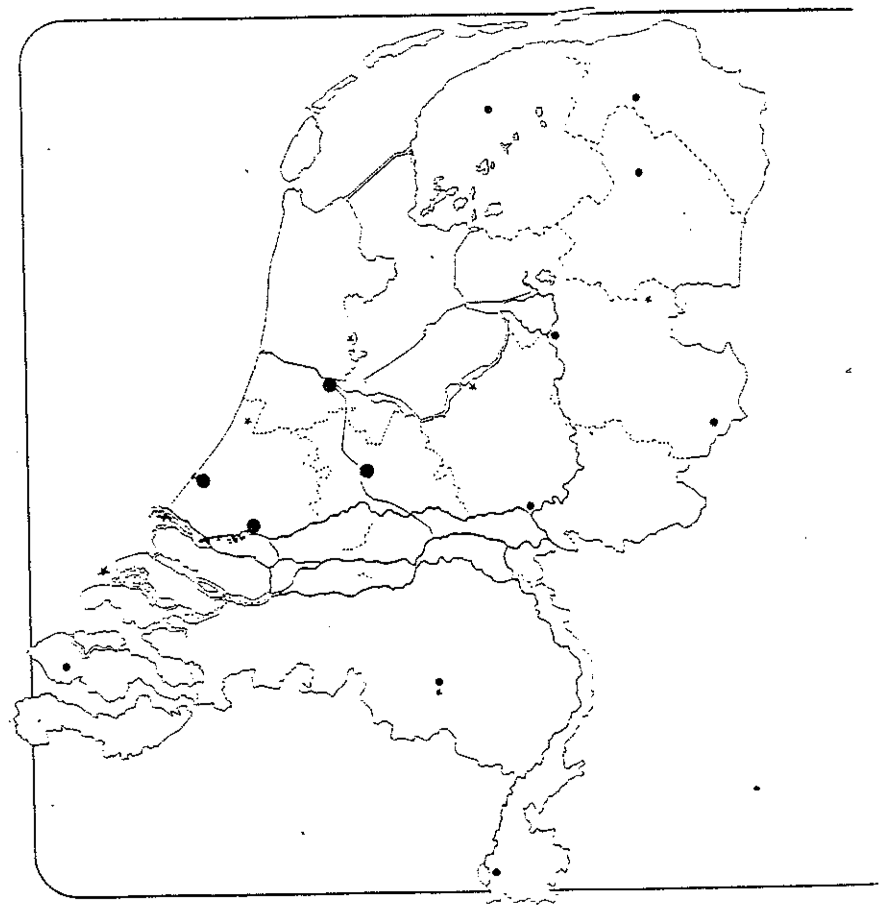
General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?



EXAMPLE 7
Simple map of the Netherlands (ITC exercise)
line-art, black & white, positive, photocopy toner,
unstable, opaque photocopy paper.

* How do you perform the scanning and vectorisation into geographic objects of small lines, wide lines, broken lines?

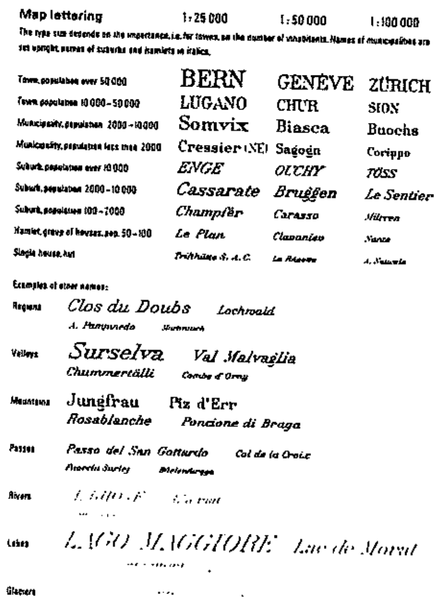
General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?



EXAMPLE 8
Map lettering (Swiss legenda)
line-art, black & white, positive, offset-print.
unstable, opaque paper.

* What problems do you envisage with this printed text?

General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?



EXAMPLE 9
Complex topographic map (Swiss Topographic Map 1:25,000)
line-art, black & white, positive, offset-print.
unstable, opaque paper.

* How could you extract contours from this image?
* Is the rock symbolisation useful in your information system?

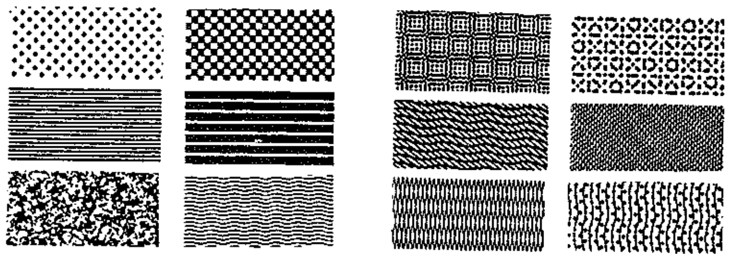
General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?

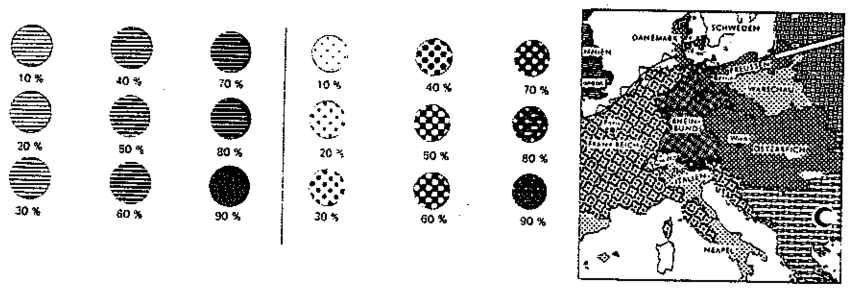
fig. 14 · Rock drawing. Reproduced from sheet 1236 of the Swiss Topographic Map 1:25,000
(REPRODUCED WITH PERMISSION FROM THE FEDERAL OFFICE OF TOPOGRAPHY FROM 15.7.1988)

EXAMPLE 10
Area symbols (ITC exercise) Patterns made up of parallel lines, curved lines, fine & coarse hatching, curves on top of dots. line-art, black & white, positive, offset-print. unstable, opaque paper.

* How do you handle these line and screen area infils? For example do you attempt to completely delete them; code them; or reproduce them?

General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?

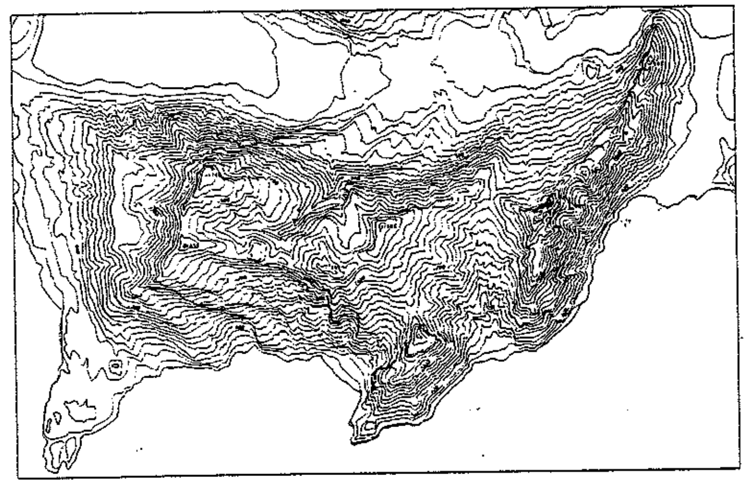




EXAMPLE 11 & 12
Film separate (contours)
line-art, black & white, photographic.
EXAMPLE 11: positive.
stable, clear film.
EXAMPLE 12: negative.
stable, clear film.

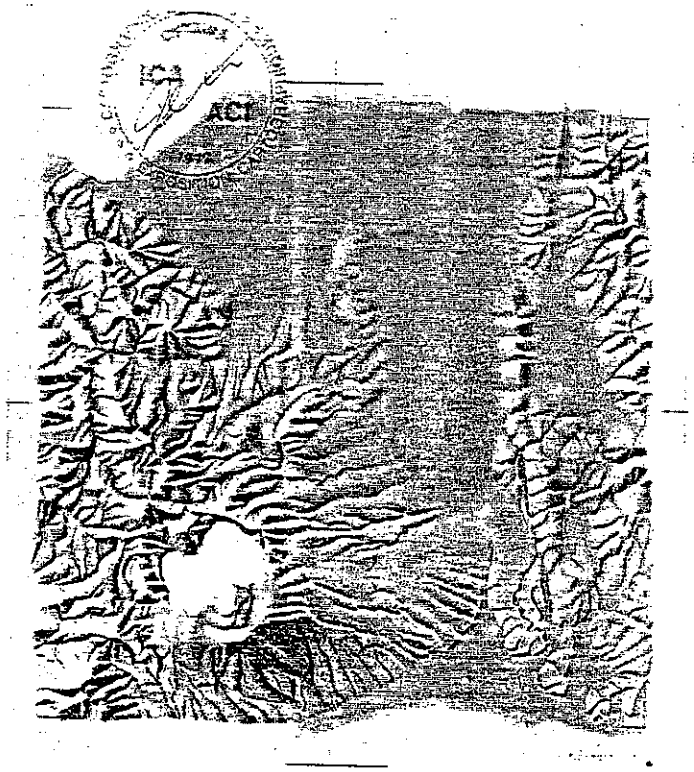
* What problems do you have with dense contouring?
* What problems do you have with handling gaps from numbering in contours?

General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?



EXAMPLE 13
Hillshading original artwork (by K. Sijmons, 1971)
continuous tone, black & white, positive, pencil & ink.
unstable, opaque paper.

General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?



EXAMPLE 14 & 15
Aerial photo
continuous tone, black & white, positive, photographic emulsion.
EXAMPLE 14: stable, clear film.
EXAMPLE 15: unstable, opaque photographic paper.

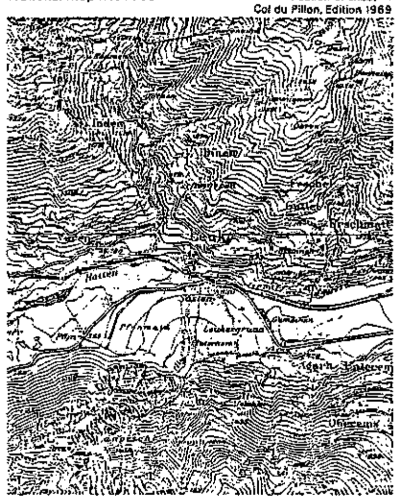
General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?

EXAMPLE 16
Complex topographic map (Swiss Topographic Map 1:100,000)
screened, colour, offset-print, positive.
unstable, opaque paper.

* What problems are introduced by the hillshading when contours are being extracted and can you cope with these?

General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?

National Map 1:100000 — Section of sheet 41 Col du Pillon, Edition 1969



Representation: With hillshading only, with or without km-grid (violet), folded or unfolded.
Size: Mapped area: 70 × 48 cm, folded map: 13 × 19 cm.
Vertical interval: 50 m — Index contours 200 m; Intermediate contours 50 m; Supplementary contours 25 m

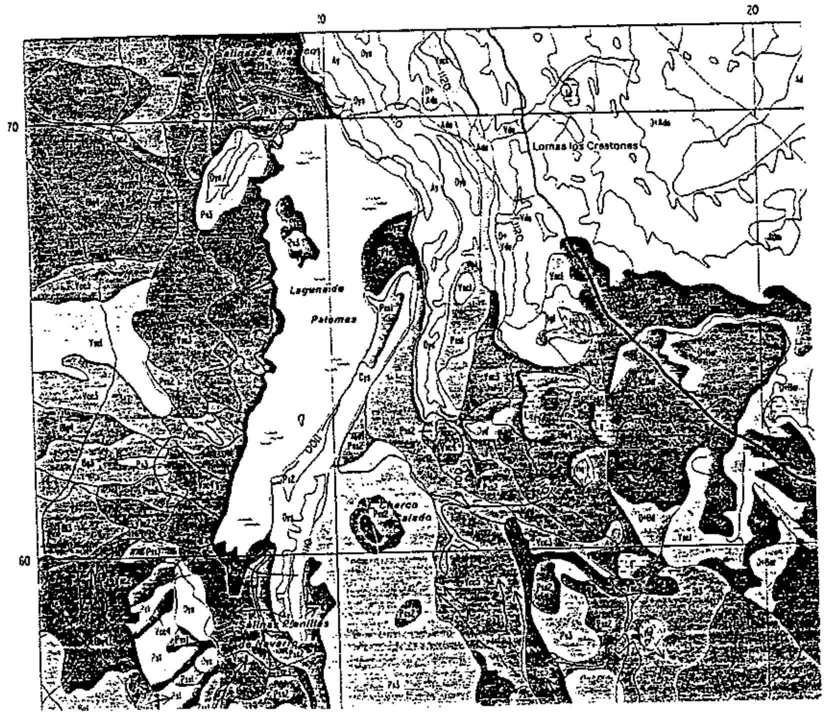
(REPRODUCED WITH THE PERMISSION OF THE FEDERAL OFFICE OF TOPOGRAPHY FROM 15.7.1988)

EXAMPLE 17
Complex thematic map (Mexico geology)
screened, colour, offset-print, positive.
unstable, opaque paper.

* What problems do you have to scan text on all coloured backgrounds?
* What problems do you have to scan contours on all coloured backgrounds?

General questions:
* Do you scan documents on such a base material?
* Do you scan such detail?
* What problems do you have when scanning such an example?
* For what do you use the resulting data?

Mapa de suelos de la reserva de la bio



DESCRIPTION OF FREEMAN CODES

The diagram below is from "Digital Picture Processing" by [Rosenfeld and Kak,1982]. In this diagram the original curved line has been simplified into its 'chain', which describes the curve by only its nearest grid intersection points. Both the curved line and its chain may be described by its Cartesian coordinates, but the chain may also be described by its FREEMAN codes. Its Freeman codes are as follows:

10112 22222 21000 01000 76645 44545 67000 012

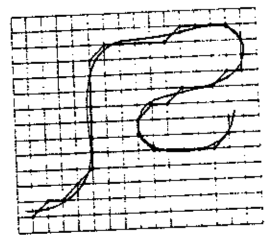


DIAGRAM 1.

Freeman codes are referred to frequently in this report. To derive Freeman codes the following guide is used:

3	2	1
4		0
5	6	7

DIAGRAM 2.

The empty central location represents the point you are at, and one of the eight surrounding locations represents the point you are going to. So considering the first point (as the point you are at) and second point (as the point you are going to) on the chain in the DIAGRAM 1, you go from the point you are at to a point in the direction 1. Then considering the second point (as the point you are at) and the third point (as the point you are going to) on the chain in the DIAGRAM 1, you go from the point you are at to a point in the direction 0. This movement from the first, to the second, to the third point...etc. is represented by the Freeman codes 10...etc.

The Freeman codes are akin to polar coordinates, with direction being a multiple of 45 degrees (Freeman code 7 implies a direction of movement 7 x 45 degrees or 315 degrees) and distance being also derived from the Freeman codes, with codes 0 and 4 representing half the x grid dimension, codes 2 and 6 half the y grid dimension, and codes 1, 3, 5, and 7 half the grid diagonal.

THE TABLES IN A GIS TO BE FILLED BY DIGITIZING

ARC TOPOLOGY

arc	From node	To node
i	Si	Ei

DOMAIN TOPOLOGY

arc	Domain on the left	Domain on the right
i	Li	Ri

NODE GEOMETRY

node	coordinates
i	X1i, X2i

ARC GEOMETRY

The shape of an arc must be known in the GIS. This is especially true if the user asks for graphic results (computer-drawn maps). Between 2 nodes, an arc can be represented by a curve in one of the following ways:

- a straight line segment;
- a polygonal line;
- an arc of a circle or ellipse; and,
- a polynomial curve.

Appendix F

THE LITERATURE DATABASE

From the literature database created using the databasing facilties of dBASEIV from Ashton-Tate, the following files have been created.

LITERATU.DBF
containing author and title of all papers

AMFMLIT.QBE
containing AMFM proceedings papers on automatic digitizing, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

AUTOCAR.QBE
AUTO-CARTO proceedings papers on automatic digitizing, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

CG&IPLIT
Jou of Computer Graphics and Image Processing, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

CHRONOLI
All literature in a chronological order, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

CONTGENL
Literature on contour generation, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

EDGEDETL
Literature on edge detection, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

FEATRECL
Literature on feature recognition, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

IEEELIT
Literature on automatic digitizing from IEEE publications, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

PATRECLI
Literature on automatic digitizing from the journal 'Pattern Recognition', with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

SKELETLI
Literature on skeletonization, with the keys author, title, year, publication details, ITC location of paper, brief summary of paper.

These files are available from Ms. Jane Drummond, Department of Cartography, ITC, P.O. Box 6, 7500 AA Enschede, The Netherlands.

LIST OF THE OEEPE PUBLICATIONS

State – June 1990

A. Official publications

1 *Trombetti, C.:* „Activité de la Commission A de l'OEEPE de 1960 à 1964" – *Cunietti, M.:* „Activité de la Commission B de l'OEEPE pendant la période septembre 1960–janvier 1964" – *Förstner, R.:* „Rapport sur les travaux et les résultats de la Commission C de l'OEEPE (1960–1964)" – *Neumaier, K.:* „Rapport de la Commission E pour Lisbonne" – *Weele, A.J. v. d.:* „Report of Commission F." – Frankfurt a. M. 1964, 50 pages with 7 tables and 9 annexes.

2 *Neumaier, K.:* „Essais d'interprétation de »Bedford« et de »Waterbury«. Rapport commun établi par les Centres de la Commission E de l'OEEPE ayant participé aux tests" – „The Interpretation Tests of »Bedford« and »Waterbury«. Common Report Established by all Participating Centres of Commission E of OEEPE" – „Essais de restitution »Bloc Suisse«. Rapport commun établi par les Centres de la Commission E de l'OEEPE ayant participé aux tests" – „Test »Schweizer Block«. Joint Report of all Centres of Commission E of OEEPE." – Frankfurt a. M. 1966, 60 pages with 44 annexes.

3 *Cunietti, M.:* „Emploi des blocs de bandes pour la cartographie à grande échelle – Résultats des recherches expérimentales organisées par la Commission B de l'O.E.E.P.E. au cours de la période 1959–1966" – „Use of Strips Connected to Blocks for Large Scale Mapping – Results of Experimental Research Organized by Commission B of the O.E.E.P.E. from 1959 through 1966." – Frankfurt a. M. 1968, 157 pages with 50 figures and 24 tables.

4 *Förstner, R.:* „Sur la précision de mesures photogrammétriques de coordonnées en terrain montagneux. Rapport sur les résultats de l'essai de Reichenbach de la Commission C de l'OEEPE" – „The Accuracy of Photogrammetric Co-ordinate Measurements in Mountainous Terrain. Report on the Results of the Reichenbach Test Commission C of the OEEPE." – Frankfurt a. M. 1968, Part I: 145 pages with 9 figures; Part II: 23 pages with 65 tables.

5 *Trombetti, C.:* „Les recherches expérimentales exécutées sur de longues bandes par la Commission A de l'OEEPE." – Frankfurt a. M. 1972, 41 pages with 1 figure, 2 tables, 96 annexes and 19 plates.

6 *Neumaier, K.:* „Essai d'interprétation. Rapports des Centres de la Commission E de l'OEEPE." – Frankfurt a. M. 1972, 38 pages with 12 tables and 5 annexes.

7 *Wiser, P.:* „Etude expérimentale de l'aérotriangulation semi-analytique. Rapport sur l'essai »Gramastetten«." – Frankfurt a. M. 1972, 36 pages with 6 figures and 8 tables.

8 „Proceedings of the OEEPE Symposium on Experimental Research on Accuracy of Aerial Triangulation (Results of Oberschwaben Tests)"
Ackermann, F.: „On Statistical Investigation into the Accuracy of Aerial Triangulation. The Test Project Oberschwaben" — „Recherches statistiques sur la précision de l'aérotriangulation. Le champ d'essai Oberschwaben" — *Belzner, H.:* „The Planning. Establishing and Flying of the Test Field Oberschwaben" — *Stark, E.:* Testblock Oberschwaben, Programme I. Results of Strip Adjustments" — *Ackermann, F.:* „Testblock Oberschwaben, Program I. Results of Block-Adjustment by Independent Models" — *Ebner, H.:* Comparison of Different Methods of Block Adjustment" — *Wiser, P.:* „Propositions pour le traitement des erreurs non-accidentelles" — *Camps, F.:* „Résultats obtenus dans le cadre du project Oberschwaben 2A" — *Cunietti, M.; Vanossi, A.:* „Etude statistique expérimentale des erreurs d'enchaînement des photogrammes" — *Kupfer, G.:* „Image Geometry as Obtained from Rheidt Test Area Photography" — *Förstner, R.:* „The Signal-Field of Baustetten. A Short Report" — *Visser, J.; Leberl, F.; Kure, J.:* „OEEPE Oberschwaben Reseau Investigations" — *Bauer, H.:* „Compensation of Systematic Errors by Analytical Block Adjustment with Common Image Deformation Parameters." — Frankfurt a. M. 1973, 350 pages with 119 figures, 68 tables and 1 annex.

9 *Beck, W.:* „The Production of Topographic Maps at 1 : 10,000 by Photogrammetric Methods. — With statistical evaluations, reproductions, style sheet and sample fragments by Landesvermessungsamt Baden-Württemberg, Stuttgart." — Frankfurt a. M. 1976, 89 pages with 10 figures, 20 tables and 20 annexes.

10 „Résultats complémentaires de l'essai d'«Oberriet» de la Commission C de l'OEEPE — Further Results of the Photogrammetric Tests of «Oberriet» of the Commission C of the OEEPE"
Härry, H.: „Mesure de points de terrain non signalisés dans le champ d'essai d'«Oberriet» — Measurements of Non-Signalized Points in the Test Field «Oberriet» (Abstract)" — *Stickler, A.; Waldhäusl, P.:* „Restitution graphique des points et des lignes non signalisés et leur comparaison avec des résultats de mesures sur le terrain dans le champ d'essai d'«Oberriet» — Graphical Plotting of Non-Signalized Points and Lines, and Comparison with Terrestrial Surveys in the Test Field «Oberriet»" — *Förstner, R.:* „Résultats complémentaires des transformations de coordonnées de l'essai d'«Oberriet» de la Commission C de l'OEEPE — Further Results from Co-ordinate Transformations of the Test «Oberriet» of Commission C of the OEEPE" — *Schürer, K.:* „Comparaison des distances d'«Oberriet» — Comparison of Distances of «Oberriet» (Abstract)." — Frankfurt a. M. 1975, 158 pages with 22 figures and 26 tables.

11 „25 années de l'OEEPE"
Verlaine, R.: „25 années d'activité de l'OEEPE" — „25 Years of OEEPE (Summary)" — *Baarda, W.:* „Mathematical Models." — Frankfurt a. M. 1979, 104 pages with 22 figures.

12 *Spiess, E.:* „Revision of 1 : 25,000 Topographic Maps by Photogrammetric Methods." — Frankfurt a. M. 1985, 228 pages with 102 figures and 30 tables.

13 *Timmerman, J.; Roos, P. A.; Schürer, K.; Förstner, R.:* On the Accuracy of Photogrammetric Measurements of Buildings — Report on the Results of the Test "Dordrecht", Carried out by Commission C of the OEEPE. — Frankfurt a. M. 1982, 144 pages with 14 figures and 36 tables.

14 *Thompson, C. N.:* Test of Digitising Methods. — Frankfurt a. M. 1984, 120 pages with 38 figures and 18 tables.

15 *Jaakkola, M.; Brindöpke, W.; Kölbl, O.; Noukka, P.:* Optimal Emulsions for Large-Scale Mapping — Test of "Steinwedel" — Commission C of the OEEPE 1981—84. — Frankfurt a. M. 1985, 102 pages with 53 figures.

16 *Waldhäusl, P.:* Results of the Vienna Test of OEEPE Commission C.— *Kölbl, O.:* Photogrammetric Versus Terrestrial Town Survey. — Frankfurt a. M. 1986, 57 pages with 16 figures, 10 tables and 7 annexes.

17 *Commission E of the OEEPE:* Influences of Reproduction Techniques on the Identification of Topographic Details on Orthophotomaps. — Frankfurt a. M. 1986, 138 pages with 51 figures, 25 tables and 6 appendices.

18 *Förstner, W.:* Final Report on the Joint Test on Gross Error Detection of OEEPE and ISP WG III/1. — Frankfurt a. M. 1986, 97 pages with 27 tables and 20 figures.

19 *Dowman, I. J.; Ducher, G.:* Spacelab Metric Camera Experiment — Test of Image Accuracy. — Frankfurt a. M. 1987, 112 pages with 13 figures, 25 tables and 7 appendices.

20 *Eichhorn, G.:* Summary of Replies to Questionnaire on Land Information Systems — Commission V — Land Information Systems. — Frankfurt a. M. 1988, 129 pages with 49 tables and 1 annex.

21 *Kölbl, O.:* Proceedings of the Workshop on Cadastral Renovation — Ecole polytechnique fédérale, Lausanne, 9—11 September, 1987. — Frankfurt a. M. 1988, 337 pages with figures, tables and appendices.

22 *Rollin, J.; Dowman, I. J.:* Map Compilation and Revision in Developing Areas — Test of Large Format Camera Imagery. — Frankfurt a. M. 1988, 35 pages with 3 figures, 9 tables and 3 appendices.

B. Special publications

— Special Publications O.E.E.P.E. — Number I

Solaini, L.; Trombetti, C.: Relation sur les travaux préliminaires de la Commission A (Triangulation aérienne aux petites et aux moyennes échelles) de l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (O.E.E.P.E.). I^ère^ Partie: Programme et organisation du travail. — *Solaini, L.; Belfiore, P.:* Travaux préliminaires de la Commission B de l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (O.E.E.P.E.) (Triangulations aériennes aux grandes échelles). — *Solaini, L.; Trombetti, C.; Belfiore, P.:* Rapport sur les travaux expérimentaux de triangulation aérienne exécutés par l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (Commission A et B). — *Lehmann, G.:* Compte rendu des travaux de la Commission C de l'O.E.E.P.E. effectués jusqu'à présent. — *Gotthardt, E.:* O.E.E.P.E. Commission C. Compte-rendu de la restitution à la Technischen Hochschule, Stuttgart, des vols d'essai du groupe I du terrain d'Oberriet. — *Brucklacher, W.:* Compte-rendu du centre «Zeiss-Aerotopograph» sur les restitutions pour la Commission C de l'O.E.E.P.E. (Restitution de la bande de vol, groupe I, vol. No. 5). — *Förstner, R.:* O.E.E.P.E. Commission C. Rapport sur la restitution effectuée dans l'Institut für Angewandte Geodäsie, Francfort sur le Main. Terrain d'essai d'Oberriet les vols No. 1 et 3 (groupe I). — I.T.C., Delft: Commission C, O.E.E.P.E. Déroulement chronologique des observations. — Photogrammetria XII (1955—1956) 3, Amsterdam 1956, pp. 79—199 with 12 figures and 11 tables.

— Publications spéciales de l'O.E.E.P.E. — Numéro II

Solaini, L.; Trombetti, C.: Relations sur les travaux préliminaires de la Commission A (Triangulation aérienne aux petites et aux moyennes échelles) de l'Organisation Européenne d'Etudes Photogrammétriques Expérimentales (O.E.E.P.E.). 2^e^ partie. Prises de vues et points de contrôle. — *Gotthardt, E.:* Rapport sur les premiers résultats de l'essai d'«Oberriet» de la Commission C de l'O.E.E.P.E. — Photogrammetria XV (1958—1959) 3, Amsterdam 1959, pp. 77—148 with 15 figures and 12 tables.

— *Trombetti, C.:* Travaux de prises de vues et préparation sur le terrain effectuées dans le 1958 sur le nouveau polygone italien pour la Commission A de l'OEEPE. — Florence 1959, 16 pages with 109 tables.

— *Trombetti, C.; Fondelli, M.:* Aérotriangulation analogique solaire. — Firence 1961, 111 pages, with 14 figures and 43 tables.

— Publications spéciales de l'O.E.E.P.E. — Numéro III

Solaini, L.; Trombetti, C.: Rapport sur les résultats des travaux d'enchaînement et de compensation exécutés pour la Commission A de l'O. E. E. P. E. jusqu'au mois de Janvier 1960. Tome 1: Tableaux et texte. Tome 2: Atlas. — Photogrammetria XVII (1960—1961) 4, Amsterdam 1961, pp. 119—326 with 69 figures and 18 tables.

— „OEEPE — Sonderveröffentlichung Nr. 1"

Gigas, E.: „Beitrag zur Geschichte der Europäischen Organisation für photogrammetrische experimentelle Untersuchungen" — *N. N.:* „Vereinbarung über die Gründung einer Europäischen Organisation für photogrammetrische experimentelle Untersuchungen" — „Zusatzprotokoll" — *Gigas, E.:* „Der Sechserausschuß" — *Brucklacher, W.:* „Kurzbericht über die Arbeiten in der Kommission A der OEEPE" — *Cunietti, M.:* „Kurzbericht des Präsidenten der Kommission B über die gegenwärtigen Versuche und Untersuchungen" — *Förstner, R.:* „Kurzbericht über die Arbeiten in der Kommission B der OEEPE" — „Kurzbericht über die Arbeiten in der Kommission C der OEEPE" — *Belzner, H.:* „Kurzbericht über die Arbeiten in der Kommission E der OEEPE" — *Schwidefsky, K.:* „Kurzbericht über die Arbeiten in der Kommission F der OEEPE" — *Meier, H.-K.:* „Kurzbericht über die Tätigkeit der Untergruppe „Numerische Verfahren" in der Kommission F der OEEPE" — *Belzner, H.:* „Versuchsfelder für internationale Versuchs- und Forschungsarbeiten." — Nachr. Kt.- u. Vermesss.-wes., R. V, Nr. 2, Frankfurt a. M. 1962, 41 pages with 3 tables and 7 annexes.

— *Rinner, K.:* Analytisch-photogrammetrische Triangulation eines Teststreifens der OEEPE. — Österr. Z. Vermess.-wes., OEEPE-Sonderveröff. Nr. 1, Wien 1962, 31 pages.

— *Neumaier, K.; Kasper, H.:* Untersuchungen zur Aerotriangulation von Überweitwinkelaufnahmen. — Österr. Z. Vermess.-wes., OEEPE-Sonderveröff. Nr. 2, Wien 1965, 4 pages with 4 annexes.

— „OEEPE — Sonderveröffentlichung Nr. 2"

Gotthardt, E.: „Erfahrungen mit analytischer Einpassung von Bildstreifen." — Nachr. Kt.- u. Vermess.-wes., R. V, Nr. 12, Frankfurt a. M. 1965, 14 pages with 2 figures and 7 tables.

— „OEEPE — Sonderveröffentlichung Nr. 3"

Neumaier, K.: „Versuch »Bedford« und »Waterbury«. Gemeinsamer Bericht aller Zentren der Kommission E der OEEPE" — „Versuch »Schweizer Block«. Gemeinsamer Bericht aller Zentren der Kommission E der OEEPE." — Nachr. Kt.- u. Vermess.-wes., R. V, Nr. 13, Frankfurt a. M. 1966, 30 pages with 44 annexes.

— *Stickler, A.; Waldhäusl, P.:* Interpretation der vorläufigen Ergebnisse der Versuche der Kommission C der OEEPE aus der Sicht des Zentrums Wien. — Österr. Z. Vermess.-wes., OEEPE-Sonderveröff. (Publ. Spéc.) Nr. 3, Wien 1967, 4 pages with 2 figures and 9 tables.

— „OEEPE — Sonderveröffentlichung Nr. 4"

Schürer, K.: „Die Höhenmeßgenauigkeit einfacher photogrammetrischer Kartiergeräte. Bemerkungen zum Versuch »Schweizer Block« der Kommission E der OEEPE." — Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M., 1968, 25 pages with 7 figures and 3 tables.

– „OEEPE – Sonderveröffentlichung Nr. 5"

Förstner, R.: „Über die Genauigkeit der photogrammetrischen Koordinatenmessung in bergigem Gelände. Bericht über die Ergebnisse des Versuchs Reichenbach der Kommission C der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1969, Part I: 74 pages with 9 figures; Part II: 65 tables.

– „OEEPE – Sonderveröffentlichung Nr. 6"

Knorr, H.: „Die Europäische Organisation für experimentelle photogrammetrische Untersuchungen – OEEPE – in den Jahren 1962 bis 1970." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1971, 44 pages with 1 figure and 3 tables.

– „OEEPE – Sonderveröffentlichung Nr. D-7"

Förstner, R.: „Das Versuchsfeld Reichenbach der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1972, 191 pages with 49 figures and 38 tables.

– „OEEPE – Sonderveröffentlichung Nr. D-8"

Neumaier, K.: „Interpretationsversuch. Berichte der Zentren der Kommission E der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1972, 33 pages with 12 tables and 5 annexes.

– „OEEPE – Sonderveröffentlichung Nr. D-9"

Beck, W.: „Herstellung topographischer Karten 1 : 10 000 auf photogrammetrischem Weg. Mit statistischen Auswertungen, Reproduktionen, Musterblatt und Kartenmustern des Landesvermessungsamts Baden-Württemberg, Stuttgart." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1976, 65 pages with 10 figures, 20 tables and 20 annexes.

– „OEEPE – Sonderveröffentlichung Nr. D-10"

Weitere Ergebnisse des Meßversuchs „Oberriet" der Kommission C der OEEPE. *Härry, H.:* „Messungen an nicht signalisierten Geländepunkten im Versuchsfeld «Oberriet»" – *Stickler, A.; Waldhäusl, P.:* „Graphische Auswertung nicht signalisierter Punkte und Linien und deren Vergleich mit Feldmessungsergebnissen im Versuchsfeld «Oberriet»" – *Förstner, R.:* „Weitere Ergebnisse aus Koordinatentransformationen des Versuchs «Oberriet» der Kommission C der OEEPE" – *Schürer, K.:* „Streckenvergleich «Oberriet»." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1975, 116 pages with 22 figures and 26 tables.

– „OEEPE – Sonderveröffentlichung Nr. D-11"

Schulz, B.-S.: „Vorschlag einer Methode zur analytischen Behandlung von Reseauaufnahmen." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1976, 34 pages with 16 tables.

– „OEEPE – Sonderveröffentlichung Nr. D-12"

Verlaine, R.: „25 Jahre OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1980, 53 pages.

– „OEEPE – Sonderveröffentlichung Nr. D-13"

Haug, G.: „Bestimmung und Korrektur systematischer Bild- und Modelldeformationen in der Aerotriangulation am Beispiel des Testfeldes „Oberschwaben." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1980, 136 pages with 25 figures and 51 tables.

– „OEEPE – Sonderveröffentlichung Nr. D–14"

Spiess, E.: „Fortführung der Topographischen Karte 1 : 25 000 mittels Photogrammetrie" (not published, see English version in OEEPE official publication No. 12)

– „OEEPE – Sonderveröffentlichung Nr. D–15"

Timmerman, J.; Roos, P. A.; Schürer, K.; Förstner, R.: „Über die Genauigkeit der photogrammetrischen Gebäudevermessung. Bericht über die Ergebnisse des Versuchs Dordrecht der Kommission C der OEEPE." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1983, 131 pages with 14 figures and 36 tables.

– „OEEPE – Sonderveröffentlichung Nr. D–16"

Kommission E der OEEPE: „Einflüsse der Reproduktionstechnik auf die Erkennbarkeit von Details in Orthophotokarten." – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1986, 130 pages with 51 figures, 25 tables and 6 annexes.

– „OEEPE – Sonderveröffentlichung Nr. D–17"

Schürer, K.: Über die Genauigkeit der Koordinaten signalisierter Punkte bei großen Bildmaßstäben. Ergebnisse des Versuchs „Wien" der Kommission C der OEEPE. – Nachr. Kt.- u. Vermess.-wes., Sonderhefte, Frankfurt a. M. 1987, 84 pages with 3 figures, 10 tables and 42 annexes.

C. Congress reports and publications in scientific reviews

- *Stickler, A.:* Interpretation of the Results of the O.E.E.P.E. Commission C. – Photogrammetria XVI (1959–1960) 1, pp. 8–12, 3 figures, 1 annexe (en langue allemande: pp. 12–16).
- *Solaini, L.; Trombetti, C.:* Results of Bridging and Adjustment Works of the Commission A of the O.E.E.P.E. from 1956 to 1959. – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 340–345, 2 tables.
- *N. N.:* Report on the Work Carried out by Commission B of the O.E.E.P.E. During the Period of September 1956–August 1960. – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 346–351, 2 tables.
- *Förstner, R.:* Bericht über die Tätigkeit und Ergebnisse der Kommission C der O.E.E.P.E. (1956–1960). – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 352–357, 1 table.
- *Bachmann, W. K.:* Essais sur la précision de la mesure des parallaxes verticales dans les appareils de restitution du I^er^ ordre. – Photogrammetria XVI (1959–1960) 4 (Spec. Congr.-No. C), pp. 358–360).
- *Wiser, P.:* Sur la reproductibilité des erreurs du cheminement aérien. – Bull. Soc. Belge Photogramm., No. 60, Juin 1960, pp. 3–11, 2 figures, 2 tables.
- *Cunietti, M.:* L'erreur de mesure des parallaxes transversales dans les appareils de restitution. – Bull. Trimestr. Soc. Belge Photogramm., No. 66, Décembre 1961, pp. 3–50, 12 figures, 22 tables.

- „OEEPE – Arbeitsberichte 1960/64 der Kommissionen A, B, C, E, F"

  Trombetti, C.: „Activité de la Commission A de l'OEEPE de 1960 à 1964" – *Cunietti, M.:* „Activité de la Commission B de l'OEEPE pendant la période septembre 1960–janvier 1964" – *Förstner, R.:* „Rapport sur les travaux et les résultats de la Commission C de l'OEEPE (1960–1964)" – *Neumaier, K.:* „Rapport de la Commission E pour Lisbonne" – *Weele, A.J. van der:* „Report of Commission F." – Nachr. Kt.- u. Vermess.-wes., R. V. Nr. 11, Frankfurt a. M. 1964, 50 pages with 7 tables and 9 annexes.
- *Cunietti, M.; Inghilleri, G.; Puliti, M.; Togliatti, G.:* Participation aux recherches sur les blocs de bandes pour la cartographie à grande échelle organisées par la Commission B de l'OEEPE. Milano, Centre CASF du Politecnico. – Boll. Geod. e Sc. affini (XXVI) 1, Firenze 1967, 104 pages.
- *Gotthardt E.:* Die Tätigkeit der Kommission B der OEEPE. – Bildmess. u. Luftbildwes. 36 (1968) 1, pp. 35–37.
- *Cunietti, M.:* Résultats des recherches expérimentales organisées par la Commission B de l'OEEPE au cours de la période 1959–1966. Résumé du Rapport final. – Présenté à l'XI^e^ Congrès International de Photogrammétrie, Lausanne 1968, Comm. III (en langues française et anglaise), 9 pages.

- *Förstner, R.:* Résumé du Rapport sur les résultats de l'essai de Reichenbach de la Commission C de l'OEEPE. – Présenté à l'XI^e^ Congrès International de Photogrammétrie, Lausanne 1968, Comm. IV (en langues française, anglaise et allemande), 28 pages, 2 figures, 2 tables.
- *Timmerman, J.:* Proef „OEEPE-Dordrecht". – ngt 74, 4. Jg., Nr. 6, Juni 1974, S. 143–154 (Kurzfassung: Versuch „OEEPE-Dordrecht". Genauigkeit photogrammetrischer Gebäudevermessung. Vorgelegt auf dem Symposium der Kommission IV der I.G.P., Paris, 24.–26. September 1974).
- *Timmerman, J.:* Report on the Commission C. "OEEPE-Dordrecht" Experiment. – Presented Paper for Comm. IV, XIII^th^ Congress of ISP, Helsinki 1976.
- *Beck, W.:* Rapport de la Commission D de l'OEEPE sur l'établissement de cartes topographiques au 1/10 000 selon le procédé photogrammétrique. – Presented Paper for Comm. IV, XIII^th^ Congress of ISP, Helsinki 1976.
- *Verlaine, R.:* La naissance et le développement de l'OEEPE – Festschrift – Dr. h. c. *Hans Härry*, 80 Jahre – Schweizerische Gesellschaft für Photogrammetrie und Wild Heerbrugg AG, Bern 1976.
- *Förstner, R.:* Internationale Versuche (Essais contrôlés) – Festschrift – Dr. h. c. *Hans Härry*, 80 Jahre. – Schweizerische Gesellschaft für Photogrammetrie und Wild Heerbrugg AG, Bern 1976.
- *Baj, E.; Cunietti, M.; Vanossi, A.:* Détermination Expérimentale des Erreurs Systématiques des Faisceaux Perspectifs. – Société Belge de Photogrammétrie, Bulletin trimestriel, Brüssel 1977, pp 21–49.
- *Timmerman, J.:* Fotogrammetrische stadskaartering de OEEPE-proef Dordrecht. – Geodesia 19, Amsterdam 1977, pp. 291–298.
- *Waldhäusl, P.:* The Vienna Experiment of the OEEPE/C. Proceedings – Standards and Specifications for Integrated Surveying and Mapping Systems. – Schriftenreihe HSBw, Heft 2, München 1978.
- *Bachmann, W. K.:* Recherches sur la stabilité des appareils de restitution photogrammétriques analogiques. – Vermessung, Photogrammetrie, Kulturtechnik, Zürich 1978, pp. 265–268.
- *Parsic, Z.:* Untersuchungen über den Einfluß signalisierter und künstlicher Verknüpfungspunkte auf die Genauigkeit der Blocktriangulation. – Vermessung, Photogrammetrie, Kulturtechnik, Zürich 1978, pp. 269–278.
- *Waldhäusl, P.:* Der Versuch Wien der OEEPE/C. – Geowissenschaftliche Mitteilungen der Studienrichtung Vermessungswesen der TU Wien, Heft 13, Wien 1978, pp. 101–124.
- *Waldhäusl, P.:* Ergebnisse des Versuches Wien der OEEPE/C. – Presented Paper for Comm. IV, XIV^th^ Congress of ISP, Hamburg 1980.
- *Timmerman, J.; Förstner, R.:* Kurzbericht über die Ergebnisse des Versuchs Dordrecht der Kommission C der OEEPE. – Presented Paper for Comm. IV, XIV^th^ Congress of ISP, Hamburg 1980.

– *Bachmann, W. K.:* Elimination des valeurs erronées dans un ensemble de mesures controlées. – Papers written in honor of the 70th birthday of Professor *Luigi Solaini* – Richerche di Geodesia Topografia e Fotogrammetria, Milano 1979, pp. 27–39.

– *Visser, J.:* The European Organisation for Experimental Photogrammetric Research (OEEPE) – The Thompson Symposium 1982. – The Photogrammetric Record, London 1982, pp. 654–668.

– *Spiess, E.:* Revision of Topographic Maps: Photogrammetric and Cartographic Methods of the Fribourg Test. – The Photogrammetric Record, London 1983, pp. 29–42.

– *Jerie, H.G. and Holland, E.W.:* Cost model project for photogrammetric processes: a progress report. – ITC Journal, Enschede 1983, pp. 154–159.

– *Ackermann, F. E.* (Editor): Seminar – Mathematical Models of Geodetic/Photogrammetric Point Determination with Regard to Outliers and Systematic Errors – Working Group III/1 of ISP – Commission A of OEEPE. – Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften, Reihe A, Heft Nr. 98, München 1983.

– *Brindöpke, W., Jaakkola, M., Noukka, P., Kölbl, O.:* Optimal Emulsions for Large Scale Mapping – OEEPE–Commission C. – Presented Paper for Comm. I, XVth Congress of ISPRS, Rio de Janeiro 1984.

– *Ackermann, F.:* Report on the Activities of Working Group III/1 During 1980–84. – Comm. III, XVth Congress of ISPRS, Rio de Janeiro 1984.

– *Förstner, W.:* Results of Test 2 on Gross Error Detection of ISP WG III/1 and OEEPE. – Comm. III, XVth Congress of ISPRS, Rio de Janeiro 1984.

– *Gros, G.:* Modèles Numériques Altimétriques – Lignes Characteristiques – OEEPE Commission B. – Comm. III, XVth Congress of ISPRS, Rio de Janeiro 1984.

– *Ducher, G.:* Preparation d'un Essai sur les Ortho- et Stereo-Orthophotos. – Comm. IV, XVth Congress of ISPRS, Rio de Janeiro 1984.

– *van Zuylen, L.:* The influence of reproduction methods on the identification of details in orthophoto maps. – ITC Journal, Enschede 1984, pp. 219–226.

– *Brindöpke, W., Jaakkola, M., Noukka, P., Kölbl, O.:* Optimale Emulsionen für großmaßstäbige Auswertungen. – Bildmess. u. Luftbildw. 53 (1985) 1, pp. 23–35.

– *van Zuylen, L.:* The influence of reproduction methods on the identification of details in orthophoto maps. – ITC Journal, Enschede 1984, pp. 219–226.

– *Visser, J.:* OEEPE-News – The European Organization for Experimental Photogrammetric Research. – ITC Journal, Enschede 1984, pp. 330–332.

– *Brindöpke, W., Jaakkola, M., Noukka, P., Kölbl, O.:* Optimale Emulsionen für großmaßstäbige Auswertungen. – Bildmess. u. Luftbildw. 53 (1985) 1, pp. 23–35.

– *Thompson, C. N.:* Some New Horizons for OEEPE. Presented Paper to the Symposium of Commission IV, ISPRS in Edinburgh, 8.–12. September 1986, pp. 299–306.

– *Dowman, I.:* The Restitution of Metric Photography Taken From Space – Comm. II, XVIth Congress of ISPRS, Kyoto 1988.

– *Kilpelä, E.:* Statistical Data on Aerial Triangulation – Comm. III, XVIth Congress of ISPRS, Kyoto 1988.

The official publications and the special publications issued in Frankfurt am Main are for sale at the

Institut für Angewandte Geodäsie
– Außenstelle Berlin –

Stauffenbergstraße 13, D-1000 Berlin 30

Organisation Européenne d'Etudes Photogrammétriques Expérimentales

Publications officielles

Content

J. Drummond (ed.): Automatic Digitizing — A Report Submitted by a Working Group of Commission D (Photogrammetry and Cartography)