tanulmányok

201/1987

MTA Számitástechnikai és Automatizálási Kutató Intézet Budapest

MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

TANULMÁNYOK A SZÁMÍTÁSTECHNIKA NYOMDAIPARI ALKALMAZÁSÁRÓL

ESZENSZKI JÓZSEF
HÉVIZI LASZLÓ
dr. KAS IVÁN
dr. LAUFER JUDIT
PALOTÁSI ANDRÁS
SZŐNYI TAMÁS
Dr. VÖRÖS KÁROLY

Tanulmányok 201/1987

A kötetben szereplő tanulmányok a 400 179 és a 420 245 számú szerződések keretében készültek.

A kiadásért felelős:

Dr. KEVICZKY LÁSZLÓ

Főosztályvezető:

Dr. CSABA LÁSZLÓ

ISBN 963 311 2419
ISSN 0237-0131

Készült a TECHNOGRAFIK - Budapest gondozásában

Készült az ORSZÁGOS MŰSZAKI INFORMÁCIÓS KÖZPONT ÉS KÖNYVTÁR házinyomdájában
(Budapest I., Gyorskocsi u. 5-7.)
Felelős vezető: Tóth Károly

E L Ő S Z Ó

Ebben a tanulmányban az integrált szövegfeldolgozás néhány kérdésével foglalkozunk.

Az első tanulmány az integrált szövegfeldolgozásból indul ki. Bemutatja a folyamat hármas tagozódását, szövegszerkesztő, adatelőkészitő és levilágitó egységekre. Körülhatárolja egy nyomdaipari adatelőkészitő egység lehetséges feladatait az automatikus fényszedő rendszereken belül és meghatározza azokat a müveleteket, amelyeket az egység hatékonyan el tud végezni. Az adott igények és lehetőségek alapján becsléseket ad az adatelőkészitő egység paramétereire, egyes részeinek felépitésére, periféria, memória igényeire stb. Végül alternativákat kinál az adatelőkészitő egység felépitésére és megvalósitására.

A következő tanulmány olyan adatelőkészitő egységgel foglalkozik, amelynek bemenete nyomdakészre szerkesztett és tördelt szöveg, mig kimenete az LG-1 laserplotterre csatlakozik. Ennek az adatelőkészitő egységnek a feladata egy speciális vektor--raszter konverzió végrehajtása az LG-1 által meghatározott idő alatt, ahol a szöveget alkotó betüket tekintjük speciális vektoroknak. A tanulmány megadja az adatelőkészitő egység felépitését, müködését, a belső adatformáit, a levilágitáshoz szükséges betükészletek adatstrukturáját.

A záró tanulmány a levilágitáshoz szükséges betükészletek számitógépes előállitásának és megjelenitésének lehetséges módjaival foglalkozik. Sorbaveszi az eljárásokat, ismerteti előnyös és hátrányos tulajdonságaikat, megfontolásokat tesz használhatóságukat illetően. Befejezésül az elkészült rendszer rövid leirását és az idő közben szerzett tapasztalatokat adja.

TARTALOMJEGYZÉK

NYOMDAIPARI GYORS ADATELŐKÉSZÍTŐ EGYSÉG NÉHÁNY KÉRDÉSE

ESZENSZKI J., HÉVIZI L., dr. KAS I.
MTA SZTAKI

BEVEZETÉS

Fényszedésen a betüknek kézi vagy gépi előállitását értjük fényképészeti úton. A végtermék többnyire film, de fotópapirra ugyancsak lehet fényszedést készitени. A fényszedő eljárások fejlettebb formája a fényszedő rendszerek, amelyek a szövegbeadásától a szövegfeldolgozáson keresztül egészen a levilágitásig többé vagy kevésbé automatikusan dolgoznak.

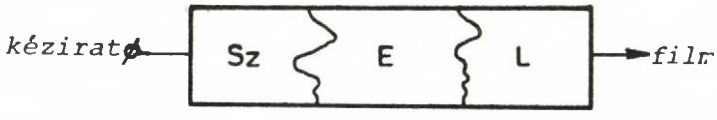
A fényszedő rendszerekben megkülönböztetünk integrált és kapcsolt rendszereket, majd ezeken belül ON és OFF LINE üzemmódokat. Integrált rendszernek azt nevezzük, amelyben a szövegrögzités, feldolgozás és levilágitás egy közös rendszerbe van foglalva és a rendszerben mindenegyes fázis csak egyszer fordul elő. Az integráltság még a külső formában is megnyilvánulhat, ha a három munkafázist megvalósitó egységek még közös burkolatot is kapnak. A kapcsolt rendszerekben az egyes fázisokat megvalósitó egységek között lazább a kapcsolat mind topológiai, fizikai, mind számbeli tekintetben [1].

1. INTEGRÁLT SZÖVEGFELDOLGOZÁS

A szövegfeldolgozás komplex feladat, amelyhez a kéziratban szereplő adathalmaztól az adathalmaz nyomtatott képének előállításáig igen sok minden tartozik. Ennek megfelelően több technikai részre bontható a feladat, hogy csak a jelentősebbeket emlitsük, a kézirat rögzitése valamilyen számitástechnikai adathordozón, a szövegszerkesztése a nyelvtani, a nyomdai és az esztétikai szabályoknak megfelelően, a levilágitás, a nyomdai lemez elkészitése.

Mégis az integrált szövegfeldolgozásnak az adatrögzitéstől a levilágitásig terjedő részét /1. ábra/ három, többé-kevésbé elkülönithető részre bonthatjuk, éspedig:

szövegszerkesztés (Sz)
adatelőkészítés (E)
levilágítás (L)



1. ábra

Integrált szövegfeldolgozás

Szövegszerkesztés (Sz) alatt a szövegfeldolgozás azon részét értjük, amely a kézirat rögzitését, szerkesztését, tördelését, korrigálását foglalja magába, egészen az ilyen módon kész szöveg rögzitéséig vagy információs csatornán történő továbbitásáig.

Az adatelőkészités (E) teremt kapcsolatot a tipográfiailag kész szöveg és a fizikai megjelenitő eszköz között. Ez az egy-

ség végzi a szövegszerkesztő egység kimenetén megjelenő viszonylag kismennyiségü, de nagy információ tartalmu adatok értelmezését, felbontását a levilágitó berendezésnek megfelelő egységekre.

A levilágitás (L) eredményezi a kézirat nyomdatechnikailag helyes képének, fotóérzékeny anyagra történő rögzitését.

Az 1. ábrán látható három egységre bontott folyamat határfelületeinél bizonyos rugalmasságot engedhetünk meg, mivel egyes feladatokat, különböző szempontok szerint, de meglehetősen szabadon sorolhatunk a szomszédos funkciók egyikébe vagy másikába.
A szemléletesség kedvéért egy pár példát emlitve:

- a szövegben felhasznált karaktereket, megfelelőképpen leirva, az adatelőkészitő egység készen kaphatja a szerkesztő egységtől, de sajátmaga is előállithatja;

- bizonyos, egyszerübb szerkesztési feladatokat - sorkizárás, oldalra, hasábra tördelés - nem fontos a szerkesztő egységnek elvégezni, rábizhatjuk az adatelőkészitő egységre is;

- a különböző pozitiv, negativ, oldalhelyes és forditott másolatok előállitásához szükséges információ transzformálások elvégzése tartozhat, mind az adatelőkészitő, mind a levilágitó egység hatáskörébe.

A következőkben a három egység feladatait részletesen tárgyaljuk. A tárgyalást a levilágitó egységgel kezdjük, tekintettel arra, hogy ez az egység készen van és paraméterei erősen befolyásolják az adatelőkészitő egység csatlakozó paramétereit.

2. LEVILÁGÍTÓ EGYSÉG (L)

A levilágitást a LASER GRAPH LG-1 tipusjelü berendezés végzi, amelyet az *MTA SZTAKI*-ban fejlesztettek ki. A berendezés alkalmas számitástechnikai eszközökkel összeállitott grafikai és szöveges információk megjelenitésére és rögzitésére fényérzékeny anyagon. Igy alkalmas nagymértékü, alakhü rajzolatok, nagy teljesitményü fényszedő rendszerekkel előállitott szövegek gyors megjelenitésére.

2.1. LG-1 SPECIFIKÁCIÓJA

Felfogható maximális film méret:	500 x 600 mm;
Tényleges rajz méret:	480 x 537,5 mm;
Programozható rasztermérét:	25 µm-es finomraszter egészszámu többszörösei;
Rajz felbontás:	25 µm;
Rajzolási idő a teljes felületre:	8 perc
Üzemmódok:	ON LINE OFF LINE
Input egységek opcionálisan:	mágnesszalag olvasó
Laser forrás:	He-Ne 5 mW-os gázlaser

2.2. LG-1 MÜKÖDÉSI ELVE

A berendezés sugárforrása egy He-Ne 5mW-os gázlaser. A sugárforrás által kibocsájtott lasernyaláb egy opto-akusztikai modulátorba megy, amely a sugárnyalábot nyolc, egymástól függetlenül modulálható, ∅ 25 µm-es nyalábra bontja. A nyolc párhuzamos sugárnyalábot mozgó-szánra szerelt optikai elemek, egy forgó dobra feszitett fényérzékeny filmre vagy papirra vetitik. A forgó dob mindenegyes körülfordulása után a mozgó szán egy scan távolságnyit továbblép a forgó dob alkotója mentén, a teljes raj-

zolási hossz megtételéig. Egy scan szélessége 8 x 25 µm, azaz 0.2 mm. A teljes rajzfelület 480 mm/0,2 mm, azaz 2400 scan.

Az LG-1 vezérlő információja, az adatmennyiség csökkentése érdekében, tömöritett formában, vagy közvetlenül, vagy mágnesesszalagon áll rendelkezésre. Az információ a berendezésben először egy 8 kByte-os belső átmeneti memóriába kerül, melynek feladata az információ átmeneti tárolása, a 16 kByte-os munka memória folyamatos információval történő ellátása érdekében. Az információ bevitel a belső átmeneti memóriába a rajzolással párhuzamosan, a rajzolási idő alatt történik. A munkamemóriában lévő információ feldolgozása Z-80 mikroprocesszszorral vezérelt cél-hardware-vel történik. A cél-hardware végzi a tömöritett információ kifejtését, a sugárkép előállitását valamint az ismétlési szám megállapitását és magának a sugár rajzolásnak a vezérlését. Két scan rajzolása között, a holtidőben történik a munkamemória feltöltése, ha erre szükség van. A belső átmeneti memóriából az információt egy DMA segitségével töltjük a munkamemóriába. A sugárvetitő optikát tartalmazó mozgószán léptetése a következő scan fölé, az uj scan-hez tartozó információ feldolgozásának megkezdése szintén a holtidőben történik. A mozgó szán léptetését, valamint a főparaméterek um. a laser intenzitás, dob fordulatszám figyelését egy másik Z-80 mikroprocesszor végzi.

Az LG-1 különféle kézi vezérlésekkel rendelkezik, amelyek közül egyesek beépitett tesztelési lehetőségeket biztositanak, mások a mozgó szán kézi vezérlését teszik lehetővé, megint mások pedig a berendezés használhatóságát terjesztik ki nagymértékben.

Igy lehetőség van PATTERN üzemben a beépitett PATTERN generátor segitségével egy tesztábrát filmre vinni, amellyel a sugárnyalábok helyes beállitását lehet ellenőrizni. Továbbá SELF üzemben a berendezés átmeneti és munkamemóriáit képes ellenőrizni.

A mozgó szán kézi vezérlése különféle beállitási lehetőségeket biztosit és a mozgó alkatrészek mechanikai ellenőrzésére ad módot.

A berendezés használhatóságát igen jelentősen kiterjeszti, hogy egy kapcsoló átállitásával ugyanazon információ bevitele mellett pozitiv vagy negativ kép rajzolható a berendezéssel és egy másik kapcsoló állitásával oldalhelyes vagy oldalforditott kép rajzolása lehetséges. Ez a két lehetőség az LG-1 használhatóságát éppen a nyomdaipari alkalmazásokban növeli meg.

A fényérzékeny anyag kazettákban kerül a berendezésbe és fel- illetőleg levitele a dobról külső kezelőszervek segitségével félautomatikusan történik. Igy a filmcsere nappali fényben mehet végbe [2].

2.3. AZ LG-1 BEMENŐ NYELVE

Az LG-1 vezérlőinformációja az átvitt adatmennyiség csökkentése érdekében tömöritve van. Ezt a tömöritett információ mennyiséget LPD-nek /Laser Plotter Data/ nevezzük és szintaxisát az alábbiakban adjuk meg. [2]

A LASER PLOTTER BEMENŐ ADATAINAK (LPD) SZINTAXISA:

<LPD>::=<FEJ><TÖRZS><FAROK>

<FEJ>::=<END OF SCAN><LÉPTÉK><END OF SCAN>

<TÖRZS>::=<SCAN/50 SOROZAT>|
<SCAN/25 SOROZAT>

<SCAN/50 SOROZAT>::=<SCAN/50>&|
*<SCAN/50><ISMÉTLŐ UTASITÁS>&|
<ÜRES UTASITÁS>&

<SCAN/25 SOROZAT>::=<SCAN/25>&|
*<SCAN/25><ISMÉTLŐ UTASITÁS>&|
<ÜRES UTASITÁS>&

<SCAN/50>::=<UTASITÁS/50 SOROZAT><END OF SCAN>

<SCAN/25>::=<UTASITÁS/25 SOROZAT><END OF SCAN>

<ISMÉTLŐ UTASITÁS>::=<ISMÉTLÉS><ADAT><END OF SCAN>

<ÜRES UTASITÁS>::=<ÜRES><ADAT><END OF SCAN>

<UTASITÁS/50 SOROZAT>::=<1BYTE-OS UTASITÁS/50>&|
<2BYTE-OS UTASITÁS/50>&|
<3BYTE-OS UTASITÁS/50>&

<UTASITÁS/25 SOROZAT>::=<2BYTE-OS UTASITÁS/25>&|
<3BYTE-OS UTASITÁS/25>&|
<4BYTE-OS UTASITÁS/25>&

<ADAT>::=<$\emptyset 1_H$> ↔ <FF_H>

<FAROK>::=<END OF PLOT>

* Megjegyzés: Ha egy SCAN memória igénye több mint 8KB-3B, akkor utána nem jöhet ismétlő utasitás!

A SZINTAXISBAN HASZNÁLT JELÖLÉSEK JELENTÉSE

<> jelek között helyezkednek el a fogalmak;

::= jel baloldalán lévő fogalmat definiáljuk a jobboldalon lévő fogalmakkal;

<><> jelek jelentik, hogy a baloldali fogalmat az adott jelekben lévő valamennyi fogalom, az adott sorrendben együttesen alkotja;

<>|<> jelek jelentik, hogy a baloldali fogalmat az adott jelekben lévő fogalmak vagylagosan alkotják;

& jel a rekurzió jelölése, a szabály baloldalán álló fogalom helyett áll a szabály jobboldalán;

↔ jel jelenti, hogy a szabály babldalán álló fogalom értéke vagylagosan, szabadon változhat a jelet közrefogó két érték között, beleértve az adott két értéket is;

CSATORNA KIOSZTÁS A SZINTAXIS EGYES FOGALMAINAK
JELENTÉSE, ÉRTÉKKÉSZLETE

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_o	
	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	<END OF SCAN>[$ØØ_H$]
	ϕ	ϕ	ϕ	1	ϕ	ϕ	ϕ	ϕ	<ISMÉTLÉS>[$1Ø_H$]
	ϕ	ϕ	1	ϕ	ϕ	ϕ	ϕ	ϕ	<ÜRES>[$2Ø_4$]
	ϕ	ϕ	1	1	ϕ	ϕ	ϕ	ϕ	<END OF PLOT>[$3Ø_H$]
	ϕ	ϕ	ϕ	ϕ	ϕ	1	ϕ	ϕ	<25μ-OS LÉPTÉK>[$ϕ4_H$]
	ϕ	ϕ	ϕ	ϕ	ϕ	1	ϕ	1	<50μ-OS LÉPTÉK>[$ϕ5_H$]
50μ-os utasitások	$S_{7,8}$	$S_{5,6}$	$S_{3,4}$	$S_{1,2}$	1	d_2	d_1	d_o	<1BYTE-OS UTASITÁS>
	$S_{7,8}$	$S_{5,6}$	$S_{3,4}$	$S_{1,2}$	ϕ	ϕ	ϕ	1	<2BYTE-OS UTASITÁS>
	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_o	
	$S_{7,8}$	$S_{5,6}$	$S_{3,4}$	$S_{1,2}$	ϕ	ϕ	1	ϕ	<3BYTE-OS UTASITÁS>
	1	d_6	d_5	d_4	d_3	d_2	d_1	d_o	
	ϕ	d_{13}	d_{12}	d_{11}	d_{10}	d_9	d_8	d_7	
25μ-os utasitások	S_4	S_3	S_2	S_1	1	d_2	d_1	d_o	<2BYTE-OS UTASITÁS>
	S_8	S_7	S_6	S_5	1	d_5	d_4	d_3	
	S_4	S_3	S_2	S_1	ϕ	ϕ	1	ϕ	<3BYTE-OS UTASITÁS>
	S_8	S_7	S_6	S_5	1	d_2	d_1	d_o	
	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	
	S_4	S_3	S_2	S_1	ϕ	ϕ	1	1	<4BYTE-OS UTASITÁS>
	S_8	S_7	S_6	S_5	1	d_2	d_1	d_o	
	1	d_9	d_8	d_7	d_6	d_5	d_4	d_3	
	d_{14}	d_{13}	d_{12}	d_{11}	d_{10}	x	x	x	

S=sugár csatorna
d=ismétlési szám
x=indiferens

A levilágitó egység bemenő felületét az információ strukturájának szempontjából ez a szintaxis adja meg. Tehát az adatelőkészitő egységnek is ebben a szintaxisban kell kiadnia az információt.

3. SZÖVEGSZERKESZTŐ EGYSÉG (Sz)

Ez az egység azokat a berendezéseket jelenti, amelyek a nyomdai folyamatláncban megelőzik az adatelőkészitő egységet (E) és a levilágitó (L) egységet. A szövegszerkesztő egység bemenete a kézirat, kimenete pedig a rendszertől függően a tartalmilag helyes, teljesen vagy részben megszerkesztett szöveg. A kimenet közvetlenül vagy adathordozón keresztül kapcsolódhat az adatelőkészitő egységre.

A fényszedési technológiák mellett az elektronikus szövegszerkesztő egység létrehozása jelentette a minőségi ugrást a nyomdaipar fejlődésében.

A szövegszerkesztő egységek központi eleme egy számitógép, melynek nagysága a hozzá kapcsolódó szerkesztő helyek számától, az operativ memória nagyságától, a vele szemben támasztott sebességi követelményektől függően lehet nagyobb vagy kisebb. A központi számitógéphez, adatrögzitő, adattároló és megjelenitő eszközök kapcsolódhatnak. A mikroprocesszorok árcsökkenésének eredményeként a számitógéphez csatolt eszközök önmagukban is rendelkezhetnek intelligenciával, ami egyuttal azt is jelenti, hogy egyre több funkciót vállalhatnak át a számitógéptől.

3.1. A SZÖVEGSZERKESZTŐ RENDSZER RÉSZEI

Az első lépés a szöveg rögzitése. Elektronikus szövegfeldolgozás esetén a digitális adathordozón történő rögzités feleslegessé teszi az anyag többszöri átgépelését. Ma már kialakult az a munkamenet, hogy a sorkizárás nélkül rögzitett szöveget egy kiiróval felszerelt mikroszámitógép sorkizárt formában kiiratja, esetleg próbalevilágitást végeznek, majd a hibák megkeresése után egy intelligens korrektura egységben dolgozzák fel, amelyben a sorképzés közben a kezelő az eredeti hibákat is kijavitja és az esetleges másodlagos hibákat is kiszüri. Igy hibátlan sorkizárt szöveget kapunk. Ehhez az szükséges, hogy a kiiratást végző számitógép és a korrektura egység a sorképzés szempontjából kompatibilis legyen. Ezt ugy alkotják meg, hogy ugyanazt a mikroszámitógép egységet használják különböző perifériákkal. Legujabban a fényszedő levilágitásba is külön sorképző processzort épitenek be.

A szövegbevivő egység billentyüzetével szemben támasztott követelmény az, hogy nagy és cserélhető karakterkészlettel rendelkezzen. A szövegbevitel display-en ellenőrizhető. Sok cég a rendszeréhez, finomfelbontásu mátrix nyomtatót is ajánl, amely alkalmas a javitott szöveg gyors és egyszerü megjelenitésére.

A központi számitógéphez ugyancsak csatlakozhat interaktiv tördelő berendezés, amely általában egy félgrafikus terminál. A rendszertől függően, az egyes képelemeket arányos méretü foltokkal, vagy a szimulált betüképpel, vagy teljesen valósághüen ábrázolja.

A fényszedő rendszerek és ezen belül a szerkesztő egységek nagy befogadó képességü és viszonylag gyors háttér memóriákat igényelnek részben a szöveg, részben a finom felbontásban ábrázolt betük számára.

A számitógépes fényszedés területén a célberendezésekre épülő rendszerek vannak többségben. Ha szerepel is általános célu számitógép központi egységként, a célperifériák, a külön-

leges kódolás és a speciális software nagyon megnehezíti a rendszerbe való belépést. A rendszerek csak arra a konfigurációra használhatók, amire megirták azokat. Az elsődleges a hardware, amelyhez a software-t mellékelik. Ma már van azonban olyan számitógépes gyártó cég is, amely maga ajánlja a teljes szövegfeldolgozó programrendszert és hardware kiépitést és vállalja ennek hozzáigazitását valamely levilágitó eszközhöz.

3.2. A SZÖVEGSZERKESZTÉS ELEMEI

Amikor a kézirat tartalmi ellenőrzése megtörtént, az anyag tipográfusa a kéziraton bejelöli a kivitelezésre szánt utasitásait.

Meghatározza a nyomtatvány alakját, szedéstükrét, a választott betütipust és betüfokozatot, az oldaljelző adatait, a beszurás mértékét, a cimrendszer kijelölését, a szöveg közti kiemelések módját, valamennyi tördelésre vonatkozó utasitást stb. Ezel képezik majd a szedés paramétereit.

A szedő program beolvassa a sorkizáratlan szöveget, majd a szedés paraméterei szerint kialakitja a nyomtatvány formáját, elvégzi a sor, hasáb és oldalképzést. Az idetartozó feladatok sokrétüek, mivel a nyomtatott szövegek igen változatosak lehetnek formájukat tekintve és külalakjukkal szemben hagyományosan igen magas követelményeket támasztanak. Ezért a sorkizárások, oldaltördelések elvégzésénél a programnak a nyelvtani szabályok mellett az esztétikai szabályokat is figyelembe kell vennie.

A bekezdések sorokra tördelésénél kivánatos, hogy minél kevesebb legyen az elválasztott szavak száma. Egymásután következő három sorban a szóközök ne egymás alá essenek. Hasonlóképpen a szöveg lapokra vagy hasábokra tördelésénél kerülendő, hogy egy bekezdés utolsó sora egy hasáb, vagy egy lap tetejére kerüljön /fattyu sor/. Az elválasztások minimalizálása csak a bekezdés teljes szövegének ismeretében oldható meg.

A szövegszerkesztéshez tartozó további feladat a megkivánt szedésminta szerinti szedés megvalósitása. Négyféle szedésmintát alkalmaznak, a jobbra-, balra zárt, sorkizárást és a tengely szimmetrikust. Az utóbbi kettőnek a megvalósitása együtt jár a szöveg nyujtásával-zsugoritásával, ami a szóközök változtatását vonja maga után.

A nyomtatott szöveg magas esztétikai minőségének elérése érdekében szükséges a betüközök változtatása is. Bizonyos betü párok egymásmellé kerülése esetén a betüket egymásra tolják, más esetekben inkább széthuzzák.

A szövegszerkesztő program eleme még a teljes vagy tört sorok pozicionálása, bekezdés és mozgó szerkesztés, táblázat szerkesztés többoszlopos szedés, cimszedés.

A gyakorlatban meglehetősen sokféle szövegszerkesztő rendszer létezik, amelyek különböző szinten oldják meg a feladatot. Vannak amelyek az itt felsorolt müveletek egyrészét intelligens perifériákra vagy adatelőkészitő egységre hagyják. Vannak azonban olyanok is, amelyek a szorosan vett szerkesztői feladatokon tul, még egyéb magasabb rendü feladatokat is ellátnak, pl. kivonatolás, tartalomjegyzék szerkesztés, gyors index, matematikai képletek egyenletek stb.

3.3. A SZÖVEGSZERKESZTŐ EGYSÉG KIMENETE

A szekesztés eredményeként létrejön a nyomtatvány képének valamilyen pontos leirása. Különböző rendszerekben ez más és más kód rendszerben, utasitás rendszerben és eltérő adathordozón jelenik meg, ami a fényszedő rendszerek épitőelemének összeférhetetlenségét okozza. Lényegük azonban közös, nevezetesen az, hogy a sima szöveg mellett rögzitésre kerül a szöveg tagolás és tipografizálás valamennyi utasitása és számértéke abszolut és/vagy relativ egységekben.

A bonyolultabb ábrák, vagy tónusos képek, amelyek nem szedhetők, vagy digitalizált formában kerülhetnek át az adatelőkészitő egységbe és azok levilágitása a szöveggel egyidőben történik vagy utólag az előhivott filmre ragaszthatók.

Az elmondottakból kitünik, hogy a számos szövegszerkesztő rendszer kimenete nem uniformizálható. Igy az adatelőkészitő egység bemenetén megjelenő információra vonatkozóan csak annyit mondhatunk, hogy szöveg, utasitás, paraméter és esetleg digitalizált kép megfelelő sorozata lehet [1], [3], [4], [5], [6], [7], [8].

4. ADATELŐKÉSZÍTŐ EGYSÉG (E)

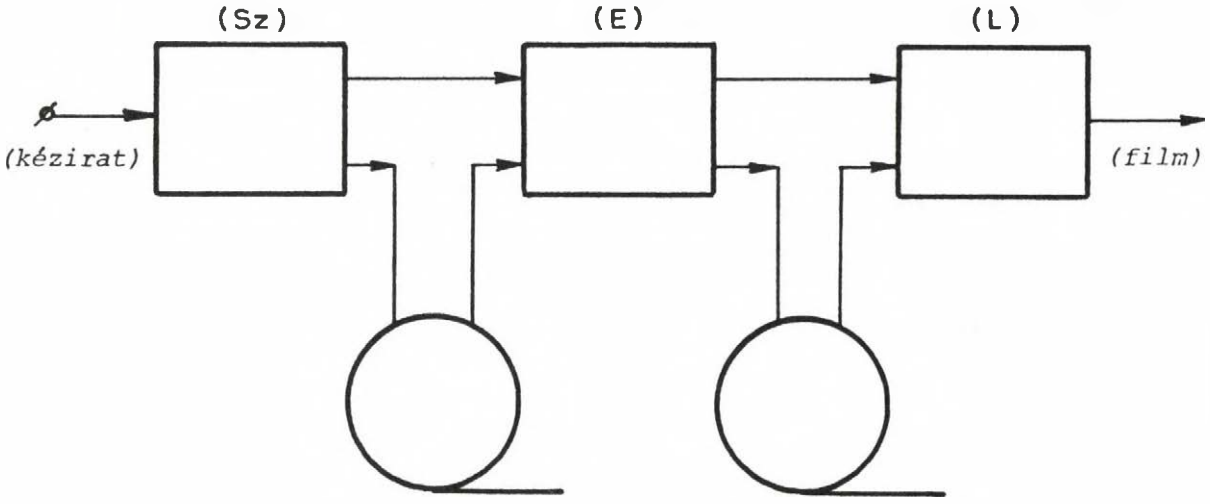
Az adatelőkészitő egység (E) állitja elő a levilágitó egységként (L) müködő LG-1 laserplotter-számára a 2. fejezetben megadott adatstrukturát. Feladata, hogy a szerkesztő egység (Sz) által megszerkesztett szöveget - ami az emberhez még közelálló, irásjelek és utasitások kombinációival adott információ - egy géporientált, meghatározott szabályokkal rendelkező adathalmazra forditsa.

Sok szakember véleménye szerint az adatelőkészitő egységre tulajdonképpen nincs is szükség, mivel rendelkezésre áll az okossá tehető és gyors számitógép, és ez meg tud oldalni mindent. Ezzel a sommás véleménnyel szemben az adatelőkészitő egység léte mellett több érv szól, igy:

- felesleges egy univerzális gépre bizni nagy mennyiségü, de viszonylag egyszerü számitástechnikai munkát, amit egy mikroprocesszoros célberendezés gyorsabban és olcsóbban elvégez;

- az adatelőkészitő egység közbeiktatása nélkül ON LINE kapcsolt szerkesztő-levilágitó rendszer kölcsönösen rontja egymás hatékonyságát;

- az adatelőkészitő egység olymódon történő közbekapcsolásával, hogy mind az (E) és mind a levilágitó egység (L) ON és OFF LINE is kapcsolható legyen /2. ábra/ növekszik a rendszer flexibilitása, a hozzáférés és a rendszer alkalmazhatósága szempontjából.



2. ábra

ON és OFF LINE kapcsolható rendszer

Mindenesetre el kell dönteni, hogy az (E) berendezést menynyi intelligenciával ruházzuk fel, hogy az alkalmazhatóság szempontjából mennyit kell a szerkesztési müveletekből az (E) berendezésre bizni.

Hasonló berendezések esetében szokásos megoldás, hogy az (E) berendezésnek két üzemmódja van. Az egyikben minden adatot készen kap az (Sz) egységtől, mig a másik esetben bizonyos utasitások, müveletek az (E) berendezésre hárulnak.

4.1. AZ ADATELŐKÉSZITŐ EGYSÉG SZINTJEI

Gyors adatelőkészitő egységgel összekötött laseres levilágitóra, gazdaságos müködés mellett a nyomdák két tipusában van szükség. Egyrészt nagy nyomdákban, ahol nagy mennyiségi igények mellett a minőségi követelmények sem elhanyagolhatók. Másrészt olyan nyomdákban, ahol kevesebb igénnyel megszerkesztett, de szintén nagy mennyiségü szövegek gyors és korszerü előállitása szükséges.

A következőkben, az eddig elmondottak alapján négy különböző szintü leirását adjuk az (E) berendezésnek.

1. A legalsó szint, a legegyszerübb (E) készülék. Ez egy nagyteljesitményü szövegfeldolgozó rendszer része. A számitógép a szövegszerkesztés folyamán teljesen alkalmazkodik a feldolgozási láncban mögötte álló levilágitást vezérlő eszközhöz. A szövegszerkesztő egység minden információt közöl az (E) egységgel. Tehát megadja a lap, a tükör méretét, hány lap, -hasáb helyezhető az (L) egység által megszabott filmre, a betütipust,-méretét, az élőfejet stb. Megadja a szöveget olyan bontásban, ahogy azt a lapok, hasábok helyzete megkivánja, olyan sorrendben ahogy az (L) egység levilágitási irányitottsága megszabja. Megadja továbbá a szövegben előforduló tipográfiai, grammatikai, esztétikai müveletekre vonatkozó utasitásokat és méreteket.

Az (E) egység csak egyetlen, az éppen felhasznált, betütipus /font/ három változatának /normál, kurziv, kövér/, egyetlen méretben történő tárolására van felkészitve.

Ez a kialakitás könyvek, ujságok kenyérbetüiből álló szövegének feldolgozására alkalmas.

2. A második szint (E) berendezése lényegileg nem különbözik az előzőtől, de komfort szintje magasabb, azaz több betükészlet tárolására alkalmas. A tárolt fontok számának és ezek méret változatainak meghatározását a nyomdai igények és

a memória kiépités költségviszonyai szabják meg. Reális igénynek 3∿5 font 3 méretben történő tárolása látszik.

Ezen felül az (E) kiépités lehetőséget biztosit szövegfeldolgozás közbeni speciális, cim vagy egyéb karakterek behivására valamilyen külső tárolóról. Természetesen ilyen esetben a karakterfeldolgozás sebessége lecsökken.

Ez a megoldás lehetővé teszi a könyvek fejezet cimeinek, ujságok fejléceinek, cimeinek automatikus feldolgozását.

3. A harmadik szinten a szövegszerkesztési munkákból az (E) egységre van bizva a szedésfajták igény szerinti megvalósitása. A berendezés tetszés szerint képes sorkizárást /tömbös szedés/, szabad soros szedést /jobbra vagy balra zárva/, középtengelyes szedést megvalósitani.

Az ilyen intelligencia foku berendezésnél természetszerüleg a nagyobb betükészlet tárolását és a külső tárolóról történő speciális karakterek behivási lehetőségét biztositani kell.

A vezérlő információ felépitésében megegyezik az első két szintével, azzal a különbséggel, hogy a szedésfajtákra vonatkozó információk mások. A szöveg tartalmazza az elválasztásokat, a sorvége jelzéseket, de a szükséges sorkiegészitéseket nem, azt az (E) egység végzi miután ismeri az adott szövegben alkalmazott szedésfajtát.

Az ebben a pontban vázolt (E) berendezés gyakorlatilag minden szöveges információ feldolgozására alkalmas olymódon, hogy a fent emlitett feladatokat átvállalja a szerkesztőegységtől. Különösen előnyös alkalmazni ujságirásnál, ahol az adatrögzitő display elött ülve elvégzi a hasábra tördelést és az elválasztásokat. Majd a szedésfajta megadása után a többit az (E) egységre bizza.

4. A negyedik szint annyiban továbbfejlesztett változata a harmadik szintnek, hogy külső adattárolóról információ feldolgozás közben lehetőség van behivni megfelelően - csatornánként, félscannenként, scannenként - tömöritett raszter képpel rendelkező tetszőleges ábrát, amit az (E) berendezés beiktat a szöveg feldolgozás folyamatába.

Ebben az esetben természetesen a bemenő információnak tartalmaznia kell a beültetendő ábra befoglaló képét meghatározó koordinátákat, megfelelő hivási cimeket stb.

A negyedik szint gyakorlatilag igy kitágitja a szövegfeldolgozást általános információ feldolgozássá.

A négy különböző szintü (E) egység lefelé kompatibilis egymással. A magasabb szinten lévők az alacsonyabb szinten lévők feladatát teljes mértékben el tudják látni. Ezzel a hierarchikus felépitéssel és moduláris kialakitással egy termék család kiépitése is elképzelhető, kielégitve igy a különböző lehetőségü felhasználók igényeit. A későbbiek folyamán, ha a szükség is megkivánja, növelni lehet a szintek számát nagyobb intelligenciával és komforttal ruházva fel az (E) egységet.

4.2. AZ ADATELŐKÉSZITŐ EGYSÉG BEMENŐ NYELVE

Az adatelőkészitő egység bemenő nyelvének kialakitásánál a legegyszerübb lenne, valamelyik Magyarországon leginkább elterjedt szövegszerkesztő egység bemenő nyelvét alkalmazni. Sajnos ez lehetetlen, mert nem beszélhetünk ilyen rendszerről.

A helyzetet még rontja, hogy számos szövegszerkesztő program létezik, de ezek kimenő nyelve még nagy vonalakban sem egyezik. /Lásd 3. fejezet./ Mivel ezeknek a programoknak a jórésze belülről nem hozzáférhető, még a kimenő nyelv és az adatelőkészitő egység bemenő nyelve közötti transzlátor program elkészitésének is kicsi az esénye. Ennek ellenére ez a módszer

ad valami lehetőséget, hogy egy meglévő szövegszerkesztő nyelvet illeszteni tudjunk az (E) egység bemenetéhez.

Ujabban foglalkoznak olyan szövegszerkesztő eljárások kialakitásával, amelyek kimenete u.n. specifikálatlan szöveg /pl. POLITEX/. Ez alatt értendő az információ olyan sorozata, amely betüket, minden rendszer által értelmezhető parancsokat és számokat, mint paramétereket tartalmaz. Ezeknek a kisérleteknek az eredményét és használhatóságát még nem lehet felmérni. Azt azonban mondhatjuk, hogy ha ez az ut eredményes lesz, akkor "specifikálatlan szöveg" és az (E) egység bemenete közé mindenféleképpen illeszthető lesz egy transzlátor.

A megoldás jelenleg az, hogy definiálunk az adatelőkészitő egység számára egy bemenő nyelvet olymódon, hogy alkalmas legyen a hierarchia bármelyik szintjén álló (E) egység kielégitésére még perspektivikusan is.

Az (E) egységgel párhuzamosan vagy később kifejlesztésre kerülő szövegszerkesztő rendszernek természetesen ezt a nyelvet mint saját kimenő nyelvét kell tekintetbe vennie.

Az adatelőkészitő egység bemenetére adott információt három részre oszthatjuk:

- szöveges üzenet, az (E) egység kezelője számára nyomtatott formában, tartalma a kezelőt érintő információ /pl. milyen betütipusokat kell beolvastatni az (E) egységgel stb./;

- az egész levilágitandó felületet érintő információ /pl. hány oldalra oszlik a film, az oldalak poziciója a filmen, tükörméret stb./;

- a feldolgozandó szöveg, amely tartalmazza a szövegre vonatkozó utasitásokat /pl. bekezdés, sorkizárás, elválasztás stb./

A szöveges üzenet fontosabb elemei:

- a betükészlet/ek/ megadása, amit az (E) egységbe előre be kell olvasni /gyémánt, garmond stb./;
- a betükészlet/ek/ méretei /4 pontos, 10 pontos stb./;
- a levilágitási forma amit az (L) egységen be kell állitani /pozitiv, oldalhelyes kép/;
- az (E) egység milyen információt /betütipus, ábra, spec. információ/ fog kérni feldolgozás közben külső adattárolóról;
- adminisztrációs adatok /azonositási szám, név, dátum stb./.

Az egész levilágitandó felületet érintő információ fontosabb részei:

- az (E) egység helyes állapotát és a megfelelő adatokkal való feltöltését végző és ellenőrző adatok, parancsok /a kezelő helyes betükészlete/ke/t, mérete/ke/t, olvastatott-e be, stb./;
- pozíciók megadására szolgáló parancsok, adatok /a filmre hány lapot, milyen pozicióba kell levilágitani, a lapon belül a tükör méret és helyzete stb./;
- a segéd adatok /adminisztrációs adatok, sorméret, sorközméret, szóközméret stb./;
- utasitás/ok/, amelyek lehetővé teszik, hogy az információs mező ezen részében adjunk meg nagyobb mennyiségü adatot /pl. még betükészletet is/.

A feldolgozandó szöveg fő elemei:

- levilágitandó szöveg karakterei;
- levilágitandó szöveg parancsai /szóközre, betüközre stb. -re vonatkozó parancsok, paraméterek/;
- perifériát kezelő közvetlen parancsok /lehetőség, külső memóriából történő közvetlen információ behivásra; közvetlen (L) utasitás stb./.

Az (E) adatelőkészitő egység bemenő nyelve az előbbiekben tárgyalt háromféle információ csoport közül csak kettőt az egész levilágitandó felületre érvényes információ "és a feldolgozandó szöveg" cimmel jelölteket tartalmazza. Ez a kétféle információ a bemenő nyelv szerkezetében is elkülönül. Az előbbit az u.n. <FEJ> tartalmazza, az utóbbit az u.n. <TÖRZS>.

A <FEJ> szerkezetileg ugy van megalkotva, hogy az oda deklarált összes parancs és adat helyet kap benne. Bővebben kifejtve, ez azt jelenti, hogy a <FEJ> számára definiálva van egy parancsokból és adatokból álló mező, amit az (E) egység értelmezni tud, de nem kötelező a feleslegesek megadása. Az (E) egység müködésében ez fennakadást nem fog okozni.

A <TÖRZS> kialakitásánál törekedni kell arra, hogy az itt megadott információ a szövegkarakterek, parancsok, paraméterek olyan sorozata legyen, ami lehetővé teszi az (E) egység számára a soros feldolgozást. Tehát azt, hogy az (E) egység által már egyszer feldolgozott területre semmilyen körülmények között ne kelljen visszanyulni.

A bemenő nyelvnek ez a szerkezeti bontása az (E) egység belső müködésében is nyomon követhető. Ugyanis a <FEJ>-ben közölt globális információt az (E) egység a tényleges szövegfeldolgozás előtt dolgozza fel. A <FEJ> feldolgozása után az (E) egység hozzá kezd a <TÖRZS> -ben megadott információ soros feldolgozásához olyan sebességgel, hogy biztositva legyen az (L) egység folyamatos mükdöése:

A bemenő információt kétféle kódolásban adhatjuk meg. Az egyik lehetséges módszer a karakteres, a másik a lineáris.

A karakter formát nevezhetjük olvasható formájunak is /ASCII kód/. Az ebben a formában leirt nyelvi attributumok, utasitások a felhasználók számára könnyebben értelmezhetők, megjelenithetők. Mivel egy font betükészlet 128 betüt tartalmaz, az utasitások számára 128 kód jut egy byte-os szavakat figyelembe véve. Ez nem feltétlenül elegendő. Ezen a hátrányon ügyes szervezéssel vagy két byte-os kódok használatával lehet segiteni.

A bináris kódrendszer elemei 1,2 vagy több byte-os formájuak lehetnek. A bináris forma tehát változó hosszuságu kód forma, amelyben az első byte a parancs. Ez határozza meg a következő byte-/ok/ értelmét és értékét. Ebben az esetben 256 utasitás megadására van lehetőség.

Természetesen a kódolási forma kiválasztásánál törekedni kell a rendszert felépitő hardware, software és firmware részek által megkivánt követelmények optimális kielégitésére, valamint az információ redundancia mentes leképzésére.

4.3. AZ ADATELŐKÉSZITŐ EGYSÉGGEL SZEMBEN TÁMASZTOTT KÖVETELMÉNYEK, TECHNIKAI ADATOK

Az adatelőkészitő (E) egységgel szemben a követelmény az, hogy első lépésben 200.000 karakter/óra feldolgozási sebessége legyen, majd technikai változtatás nélkül 800.000 karakter/óra sebességet lehessen vele elérni. A perspektivikus sebesség pedig 2.10^6 karakter/óra. Ezzel a jelen tanulmányban nem foglalkozunk.

A levilágitást vezérlő (E) egységnek az (L) laserplottert ugy kell táplálni bemenő adattal, hogy annak müködése folyamatos legyen. Ez kemény feltételt támaszt a hardware és a firmware kialakitásávan szemben. Rövid idő alatt óriási adatmeny-

nyiséget kell ahhoz feldolgozni, hogy a kivánt 200.000 karakter/óra teljesitményt a berendezés elérje.

A ma elérhető csucsteljesitményt a levilágitó egységként müködő laserplotter (L) sebessége határozza meg. A laserplotter (L) jelenleg 480 x 530 mm-es azaz 254100 mm^2-es film felületet képes nyolc perc alatt levilágitani. Teljes kitöltést feltételezve, ekkora filmen, a legkisebb gyémánt betüből aminek a méretei 1,5 x 1 mm azaz 1,5 mm^2 170.000 db, a körülbelül 10 mm^2-es garmond tipusból pedig 25000 darab fér el. Ez alapján megbecsülhetjük az elérhető csucs teljesitményt egy munka órára vetitve. /1. táblázat/.

Betütipus	Méret pont	Méret mm^2	1 levilágitás alatt db	1 óra alatt
Gyémánt	4	1.5 mm^2	170.000	1.300.000
Kolonel	7	4.6 mm^2	55.000	418.000
Garmond	10	10 mm^2	25.000	187.500
Mittel	14	18.4 mm^2	14.000	105.000

A laserplotter által elérhető teljesitmény különböző betütipus esetén.

1. táblázat

Az (E) egység szempontjából is meg kell vizsgálni, hogy mit jelentenek ezek a teljesitmények az időzités szempontjából. A laserplotter egy scan-t 200 msec alatt rajzol le. Ha a szöveg soriránya megegyezik a scan iránnyal és ha az időket tekintve a legrosszabb esetet vesszük figyelembe, azt, hogy a sorban gyémánt betük vannak, akkor a laserplotter folyamatos táplálását feltételezve a rajzolandó scan-kép előállitása közben 300 µsec áll rendelkezésre egy-egy betü egy scan szeletének a feldolgozására.

A betütipus magasságától függően a szövegsorok feldolgozására milliméterenként egy szekundum áll rendelkezésre. A különböző betütipusokat alapul véve a feldolgozási időket a 2. táblázat mutatja.

Betütipus	Betü magasság	Betü szélesség	Egy betü, egy scan feldolgozási ideje	Egy nyomtatott sor feldolgozási ideje
			sec	sec
Gyémánt	1.5 mm	∿ 1 mm	300	1.5
Kolonel	2.63 mm	∿ 1.8 mm	540	2.63
Garmond	3.7 mm	∿ 2.5 mm	750	3.7
Mittel	5.26 mm	∿ 3.5	1050	5.26

A különböző betütipusok feldolgozási ideje.

2. táblázat

Ezek elegendően nagy időknek látszanak ahhoz, hogy az (E) egység párhuzamos feldolgozást feltételezve - egyszerübb szerkesztési, vagy már kevésbé gépies müveleteket hajtson végre a bemeneti adathalmazon a tényleges adatátalakitás mellett.

Lehetőség van még bizonyos korlátozott mértékü sebesség növelésére a laserplotter sugárforrás intenzitásának 10∿15 %-os növelésével és a filmhordozó forgódob fordulatszámának ∿10 %-os növelésével. Perspektivikusab, ennél nagyobb sebesség elérése a laserplotter áttervezését vonja maga után, nevezetesen 16 bites, nagyobb sebességü processzorokkal és gyorsabb egyéb áramköri elemekkel.

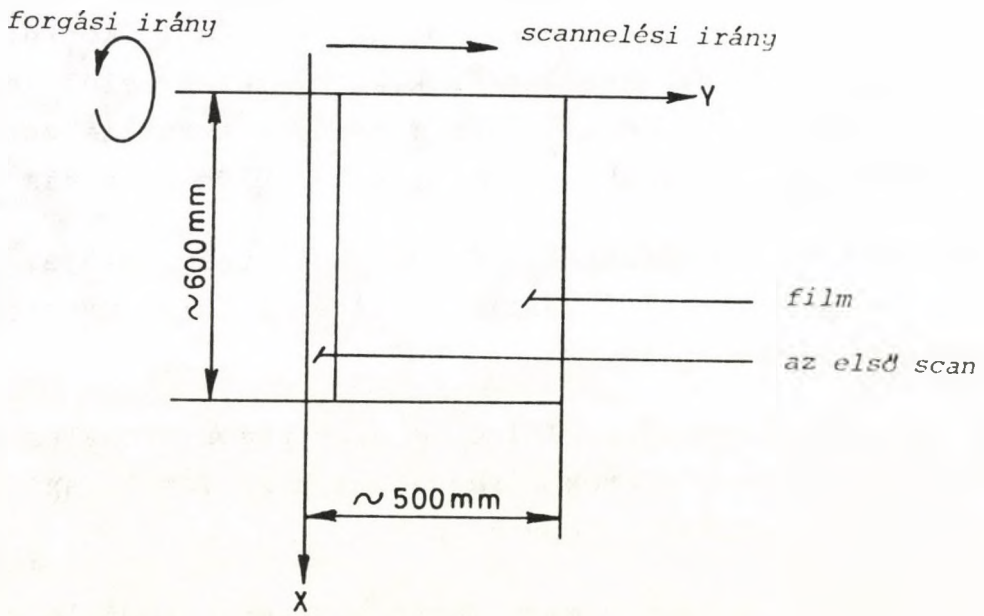
4.4. POZICIONÁLÁS

Pozicionálás alatt a levilágitandó szöveg elhelyezését értjük a filmen. Ennek a kérdésnek az elemzésénél két adott tényt kell figyelembe venni. Egyrészt a nyomdaiparban használatos léptéket, másrészt a laserplotter által nyujtott lehetőségeket.

A nyomdaiparban szokásosan három különböző finomságu léptéket alkalmaznak:

- karakteren belül 25 µm
- soron belül 0,1 mm
- oldalon belül 2 mm

A laserplotter rajzolási finomsága 25 µm, koordináta rendszerének irányitottságát pedig a 3. ábra mutatja.



3. ábra

A laserplotter koordináta rendszerének irányitottsága

A szöveg olvashatósága szempontjából a szöveget elvileg két irányba vihetjük fel a filmre, scan-irányban és arra merőlegesen. Az (E) egységben alkalmazott iránynak a kiválasztása az egyik legdöntőbb kérdés. Ezt csak az (E) egységre bizott konkrét feladatok processzálások és szerkesztési feladatok ismeretében és az azokat megvalósitó algoritmusok kidolgozása után lehet felelősségteljesen megmondani. Előzetes becsléseket végezve a scan iránnyal megegyező olvashatóságot tartjuk kivánatosnak, elsősorban abban az esetben, ha bizonyos szerkesztési müveleteket /sorkizárás/ is végez az (E) egység.

A levilágitó film mérete olyan nagyságu, hogy valószinüleg egyszerre több hasáb vagy több lap kerül egy filmre. A lap vagy hasáb pozicionálása feltétlenül az (E) egység bemenő adataiként a <FEJ>-ben található.

A poziciók megadása Descartes koordináta rendszerben történik. A pozicionáló koordinátának két tipusa van: abszolut és relativ. Az abszolut koordináták az elsődlegesek és a teljes exponálandó felületre vonatkoznak. A relativ koordináták a másodlagosak és az aktuális szük környezeten belül jelntenek pozicionálást. A koordináták mindig az aktuális léptékben vannak megadva. A lépték változtatására külön utasitás szolgál.

Az abszolut koordináták 2-2 byte hosszuságu előjel nélküli számok, amelyek lehetővé teszik a teljes felület megcimzését a legfinomabb 25 µm-es léptékben.

A relativ koordináták 1-1 byte hosszuságu előjeles számértékek, igy 25 µm-es léptékben az adott poziciótól maximálisan ±3.175 mm-es távolságot képviselnek.

A karakter poziciója megoldható elvileg háromféle képpen:

- a karakter bázis vonalának bal végénél;
- a teljes karakter befoglaló formájának bal alsó sarkánál;
- a teljes karakter befoglaló formájának bal felső sarkánál;

A nyomdatechnikai alkalmazásokhoz a bázisvonalas megadás áll a legközelebb, de az exponáláskor a karakter bázis vonalát át kell számitani a karakter kezdő scan-jeinek pozíciójára.

Az exponálás irányitottsága miatt a bal felső sarokban történő megadás igényli a legkevesebb átszámitás elvégzését.

4.5. BETÜKÉSZLET MEGADÁSA

A betükészlet meghatározása az információs rendszerben egy adott kóddal történik és lényegében szabadon választható. Követelmény a kód rendszerrel szemben az, hogy a használt betütipus fajták, változatok és méret sorok leirhatók legyenek vele és az (Sz) és (E) egységek ugyanazon kód alatt ugyanazt a betükészletet értsék.

Betükészlet bevitelkor, illetőleg tárolásakor a következő adatokat kell megadni:

- a betü hivási kódját, betükészlet esetén csak egy kitüntetett betü kódját szükséges megadni a többi inkrementálisan megadható;
- a betü méretét 25 µm-ben, fél vagy egész scan-ben;
- a betü képét.

A betükép tárolására több lehetőség van:

- teljes raszterkép;
- csatornára tömöritett raszter kép;
- fél scan-re tömöritett raszter kép;
- scan-re tömöritett raszter kép.

A 3. táblázatban bemutatjuk egy garmond m betü esetében milyen tároló kapacitásra van szükség a teljes raszter képhez viszonyitva a különböző tömöritési eljárások mellett.

Tömörités módja	Byte	Tömöritett raszter $\xi = \frac{\text{tömöritett}}{\text{teljes}} \cdot 100$	Megjegyzés
Teljes raszter-kép	1.7 kByte	100%	A betü képtől nem, csak a betü nagyságtól függő tárolási forma
* Csatorna tömörités	536 Byte	33%	A scan irány merőleges a betü állására
	360 Byte	20%	A scan irány megegyezik a betü állásával
Fél scan	168 Byte	∿10%	A scan irány merőleges a betü állására
	126 Byte	∿7%	A scan irány megegyezik a betü állásával
Scan	342 Byte	∿19%	A scan irány merőleges a betü állására
	242 Byte	∿13%	A scan irány megegyezik a betü állásával

* *Csatorna alatt egy scan egyik sugarát értjük.*

A különféle tömöritések hatása a tárolási kapacitásra garmondra m betü esetén.

3. táblázat

Ez a táblázat elsősorban arra jó, hogy a különféle tömöritések memória kapacitásra gyakorolt hatását megmutassa. Meszszemenő következtetést ebből nem szabad levonni - tulajdonképpen több betüből álló statisztikára lenne szükség - azonban azt megállapithatjuk, hogy a fél scan-re tömöritett forma a legkedvezőbb, ezen belül pedig az ahol a scan irány egybe-

esik a betüállással. Ezt az előnyt elveszitjük, ha a scannelési irány erre merőleges. Megállapithatjuk még, hogy a tömöritett forma memória igénye függ a betü képétől és nagyságától is.

Adattömöritésre a fentieken kivül használható lenne az ismétlődő minták, formák felhasználása. Ennek hátránya, hogy nem veszi figyelembe a csatorna ill. scan orientáltságot és a scan kép összeállitásánál okozna felesleges időveszteséget.

Memória megtakaritást okozna esetleg még az optimálishoz közeli kódolás megvalósitása, de ez a rendszer is veszit előnyéből, ha a processzálási időket vesszük alapul.

Egy nyomtatott szövegben a betüknek és a belőlük alkotott soroknak a mozgás szabadsága 0.1 mm és ez megfelel éppen fél scan-nek, a további feldolgozást tekintve a leggazdaságosabbnak látszik, mind a memória igényt, mind az algoritmus időt tekintve a fél scan-re tömörités.

Egy tetszőleges szövegben előforduló betük döntő többségét egy betütipus, annak esetleg még két változata, egy méretben, teszi ki. Például garmond /10 pont, normál, kövér, kurziv/. Egyetlen ilyen kollekció tárolása, fél scan-es tömöritésben 3 x 128 x 200 Byte-ot, azaz 75 kByte-ot jelent 10 pontos betüméret esetén.

Amennyiben megelégszünk három készlet vagy egy készlet három méretben történő tárolásával, akkor egy 256 kByte-os félvezető memória erre elegendő.

Ha ez kevésnek bizonyulna, akkor több lehetőség közül választhatunk:

- növeljük a félvezető memória kapacitást, de mindig lesz olyan betü készlet, ami éppen nem fér bele;
- az (Sz) program külön megadja és előállitja a szövegben előforduló összes betü tömöritett képét és csak ezt tá-

roljuk belül. A memória kapacitás valószinüleg megmaradna a 256 kByte-os nagyságban.

- a betükészletek egy méretben vannak tárolva, a méret sorozatot generátor függvények állitják elő. /LASERCOMP/. A rendszernek gyorsan jó minőségben kell tudni nagyitani és/vagy kicsinyiteni.

- a betükészletek verzióit generátor függvény állitja elő. A rendszernek gyorsan és jó minőségben kell a betüket transzformálni.

Véleményünk szerint első lépésben egy 256 kByte-os félvezető memória betükészlet/ek/ tárolására elegendőnek látszik.

A betükészlet bevitele a belső memóriába két csatornán történhet:

- külön beviteli vonalról;
- a fő adatbeviteli vonalról.

A külön beviteli vonalhoz egy tárolóegység /célszerüen floppy disk/ csatlakozik. A csatoló viszonylag gyors adatbevitelt tesz lehetővé.

A fő adatbeviteli vonalról is lehet bevinni karaktereket /esetleg teljes készletet/, de mivel ez a feldolgozás sebességét csökkenti, célszerübb nem vagy csak korlátozottan alkalmazni. [1],[2],[10].

4.6. AZ ADATELŐKÉSZÍTŐ EGYSÉG FELÉPÍTÉSE

Ebben a fejezetben az adatelőkészitő (E) egység vázlatos felépitését adjuk meg figyelembe véve az előzőekbentett megfontolásokat és megkötéseket. A fejezet röviden megadja az egyes funkcionális blokkok feladatait, becsléseket tesz az egyes egységek memória és idő viszonyaira, de nem tárgyalja azok megvalósitásait és algoritmusait.

A 4. ábrán a nyomdaipari gyors adatelőkészitő (E) egység bemenő- és kimenő perifériái valamint magának az (E) egységnek a felépitése látható. Az ábrán a négy különböző szintü (E) egység közös blokk vázlatát ábrázoltuk olymódon, hogy az alapszintü berendezéshez tartozó egységek folytonos vonallal vannak ábrázolva, mig a járulékos berendezések szaggatott vonallal. A blokk vázlat csak az információ áramlást jelzi és nem tartalmazza a vezérlő jelek kapcsolatát.

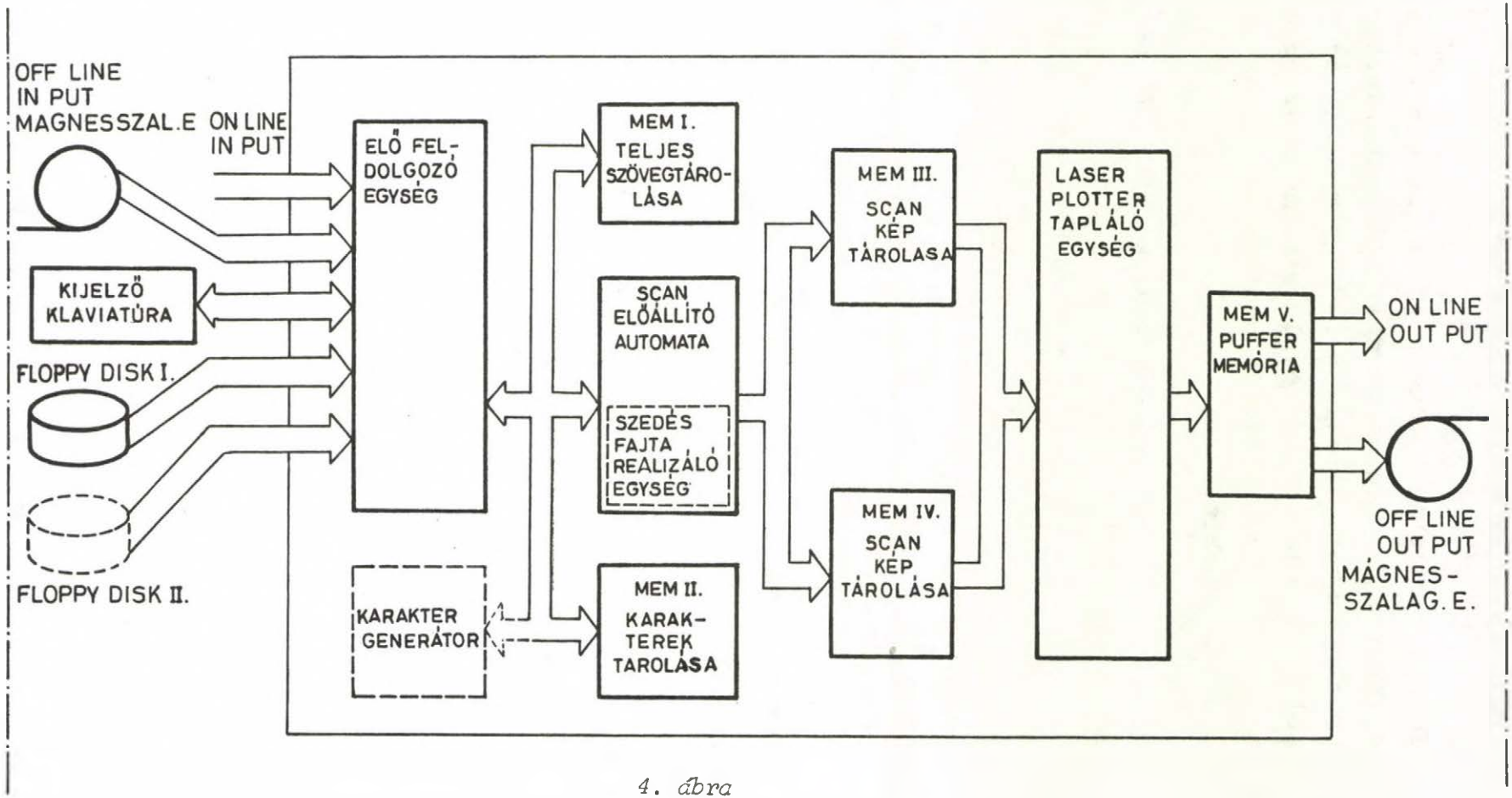
1. Bemenő perifériák:

   - ON LINE összeköttetés az (Sz) egységgel. Ez ad lehetőséget az (Sz) egységnek, hogy az (E) egységbe közvetlenül táplálja be a feldolgozni kivánt információt.

   - MÁGNESSZALAGOS egység biztositja az OFF LINE összeköttetést, tartalmazza a kivánt formában a feldolgozásra kerülő szöveget.

   - FLOPPY DISK I bemenet biztositja a belső tárolásra kerülő "kenyér" betükészletet vagy készleteket.

   - FLOPPY DISK II a magasabb szintü (E) egységeknél a speciális karakterek vagy egyéb információk külső tárolására szolgál. Az (E) berendezés a levilágitásra kerülő szöveg feldolgozása közben fordul ehhez a perifériához.



4. ábra

A nyomdaipari gyors adatelőkészitő (E) egység blokk vázlata

- KIJELZŐ és KLAVIATURA az igényektől függő egyszerübb vagy bonyolultabb kezelő-gép információ cserét tesz lehetővé.

2. Előfeldolgozó egység

Ennek az egységnek a feladata több rétegü. Egyrészt neki kell biztositani az (E) egység számára, a kapcsolatot a külvilággal. Az (Sz) egység adatai ide érkeznek be a különféle perifériális eszközökből és magából az (Sz) egységből. Másrészt a megkapott információt ez az egység osztja szét és továbbitja az arra illetékes többi részegység számára. Harmadrészt, az (E) egységgel szemben támasztott követelményektől függően különböző globális információ feldolgozást végez.

3. Karakter generátor

Igénytől függően akkor van rá szükség, ha az egyes betük generátor függvénnyel vannak megadva és az (E) egységnek kell a betük képeit ezekből előállitania. Szokásos megoldásaiknál a betük generálása ∿100 µsec alatt megy végbe. Ez az idő a berendezéssel szemben támasztott követelményekkel nincs ellentétben.

4. MEM I.

Egyrészt az előfeldolgozó egység munkamóriája, másrészt a levilágitandó filmre helyezhető teljes szöveg tárolását kell biztositania. Az ebben szereplő szöveg alapján a levilágitás vezérelhető, azaz a levilágitás sorrendje szerint vannak az adatok elhelyezve benne.

Nagyságára nézve a következő becslés adható:

- az egész filmre folyamatosan a legkisebb betü tipusból álló szöveget visszük fel;
- ez a betütipus a gyémánt, aminek a befoglaló méretei 1.5 x ∿1 mm ≃ 1.5 mm^2;
- a teljes film felület 480 x 530 mm = 254 400 mm^2;
- a teljes film felületre 254 400/1.5 = 169600 darab betü fér fel;
- egy betünek a tárolásához 1 byte kell pl. ASC II módban akkor ez 169 600 byte azaz felfelé kerekitve 166 kByte.

Tehát egy 256 kByte-os memória megfelelő tartalékkal elegendően nagy a teljes film felület levilágitására szolgáló szöveg tárolására. A tartalék 90 kByte-os memória kapacitásnak a szövegben előforduló utasitások tárolására szintén elegendőnek kell lenni. Amennyiben ez még sem lenne elegendő, a berendezés ebben a speciális és igen ritkán előforduló esetben csak a tárolt szövegnek /256 kByte/ megfelelő film méretet fogja levilágitani. Ennek ellenőrzése és jelzése az előfeldolgozó egység feladata.

A memória tipusa megfelelően megválasztott RAM;

5. MEM II.

A levilágitásra kerülő karakterek digitálisan, tömöritve tárolt képek kerülnek ebbe a memóriába. A MEM II. gyorsan szolgáltatja a scan előállitó automata számára az éppen elhelyezésre kerülő betü megfelelő scan szeletét.

Nagyságára nézve a 4.5. fejezetben adtunk becslést, aminek az értéke 256 kByte.

Tipusa megfelelően gyors elérési és cimezési idejü dinamikus RAM.

6. Scan előállitó automata

Ez az egység végzi a scan kép előállitását. Bemenő adatait egyrészt a MEM I-ből - a szöveget -, másrészt a MEM II-ből - a szöveget alkotó betüket - kapja. Kimenő memóriái a MEM III és MEM IV. a MEM III. és IV-ben kerülnek tárolásra a folyamatosan készülő scan képek.

A scan előállitó automata müködési sebességének olyan nagynak kell lenni, hogy a levilágitó (L) egység vezérlő adatokkal történő ellátása folyamatos legyen. Mivel egy scan-t a levilágitó 200 msec alatt világit le, ezért a scan kép előállitására is maximálisan ennyi idő áll rendelkezésre. Ez az idő korlát az automatára nagy feladatot ró. Hagyományos mikroprocesszoros technikával ez valószinüleg nem oldható meg.

A szóbajöhető megoldások:

- bit-slice processzoros megoldás;
- Z-8000-es gyorsabb kivitelü processzor;
- Z-8000-es normál processzorokból felépitett PIPELINE technika;
- Z-8000-es nromál processzorokból kialakitott paralel feldolgozó rendszer;
- Z-8000-es normál processzorral vezérelt programozható nagy sebességü cél-hardware.

Anélkül, hogy a lehetséges megoldások tárgyalásába bocsájtkoznánk, megjegyezzük, hogy a MEM III és MEM IV memóriák bizonyos párhuzamosan megvalósitható müveletek elvégzésére adnak

lehetőséget.

A scan előállitó automata, hardware-jén kivül, másik fontos eleme a jól megválasztott algoritmus. Az algoritmussal szemben támasztott követelmény a sebesség mellett az, hogy a scan képet a beérkező adatokból lehetőleg bitsorozatokkal végzett gyors alapmüveletekkel /shiftelés, or, and, komparálás/ tudja összeállitani.

A scan előállitó automata feladata még, magasabb követelményeket kielégitő (E) berendezés esetén, a szedés fajták megvalósitása. A rendszernek itt bizonyos számolási és helykiosztási és paraméter módositási müveleteket is kell végezni. Ennek a problémának a megoldását is elősegiti a párhuzamos feldolgozási hierarchia.

7. MEM III és MEM IV.

Az előnyösnek itélt párhuzamos feldolgozás érdekében különböztetünk meg MEM III. és MEM IV-s egységeket. A két egység tehát mind mennyiségileg, mind minőségileg megegyezik egymással.

A scan előállitó automata tölti fel azokat a scan képpel. Egy scan teljes leirását tartalmazzák, de nem feltétlenül a végleges formában. Amikor egy scan kép előállitása befejeződik a scan előállitó automata a másik memóriára vált.

A memória nagysága a bemenő információtól, a scan képet előállitó algoritmustól függ. Ha az algoritmus olyan, hogy

- egy teljes scan-t tömöritès nélküli formában állit össze, akkor mivel egy scan 21504 raszterből áll és egy scan nyolc sugárból, akkor a memória szükséglet 21504/1024 $\simeq$ 21 kByte;

- egy teljes scan tömöritett képét állitja össze, akkor a becslés gondolatmenete a következő:

  - a sugárkép tárolásához kell 1 byte;
  - maximálisan 256 lehet az ismétlési szám akkor, ehhez szintén 1 byte szükséges;
  - a tömörités mértéke jobb 50%-nál, akkor az össz memória igény megegyezik az előzővel azaz 21 kbyte;
- egy fél scan tömöritett képét állitja össze, majd a két fél scan-ből állitja össze az egész scan-t, akkor a memória szükséglet megegyezik az előzővel. A becslés gondolatmenete megegyezik az előzővel.

A becsléseket a tömöritésre vonatkozóan megengedhetjük, ha figyelembe vesszük az 1. táblázat azon adatait, amelyek a gyémánt betü scan-re és fél scan-re tömöritett byte szükségleteit adják. A gyémánt betü a tömörités szempontjából a legrosszabb.

Amint látjuk a háromféle megoldás azonos memória szükséglet eredményez. A három megoldást megvalósitó mindegyik algoritmusnak meg van a maga előnye és hátránya.

A memória szükséglet tehát 21 kByte. Mivel két memória egységünk van MEM III és MEM IV ez 2 x 21 kByte azaz 42 kByte memória kapacitást jelent.

A memória tipusa itt is dinamikus RAM.

8. Laserplotter tápláló egység

Az éppen felszabadult RAM III vagy RAM IV tartalmát tölti, a kimeneti puffer memóriába. Ha a feldolgozás módja megkivánja ez az egység állitja elő a scan képből a laserplotter által megkivánt formába a kimeneti adatokat. Egyuttal ez az egység vezérli a kommunikációt a kimenő oldali külvilággal.

9. MEM V.

A kimeneti puffer memória, tartalmazza az LG-1 egység bemenetére ON vagy OFF LINE uton jutó információt. Az információ formájának meg kell felelni az LG-1 bemenő szintaxisának. /2.3. fejezet/.

A memória nagyságát megszabja az ON LINE üzemben vele szemben álló LG-1 levilágitó egység belső átmeneti memória kapacitása, ami 8 kByte. OFF LINE üzemben a mágnesszalagos egység blokk hosszusága szabja meg a puffer memória nagyságát. Jelen esetben az kisebb mint 8 kByte, igy ez fennakadást nem okoz.

A memória tipusa dinamikus RAM.

10. Kimenő perifériák

- ON LINE összeköttetés közvetlenül az (L) egység bemenetére csatlakozva ezen keresztül a levilágitó egység folyamatos vezérlése lehetséges;

- MÁGNESSZALAGOS egység biztositja az OFF LINE összeköttetést, tartalmazza a kivánt formában a levilágitó egység vezérlő információját.

Ö S S Z E F O G L A L Á S

A tanulmányban körülhatároltuk egy nyomdaipari gyors adatelőkészitő egység lehetséges feladatait, meghatároztuk azokat a müveleteket, amelyeket az egységnek el kell végezni és amit még hatékonyan el tud végezni. Definiáltuk az egység be- és kimenő felületeit. Megadtuk az egység teljesitmény karakterisztikáját a levilágitó egységgel összekapcsolva, négy betütipus függvényében. Javaslatot tettünk egy hierarchikus modulárisan felépithető termék család kifejlesztésére, amelynek megadtuk a belső logikai felépitését, be- és kimenő perifériáit, becslést adtunk belső memória igényeire.

Mivel nem tartozott a kitüzött célok közé, a tanulmányban nem foglalkoztunk a berendezés konkrét megvalósitásával, valamint az információt feldolgozó algoritmusokkal.

Megállapithatjuk, hogy egy ilyen tipusu adatelőkészitő berendezés-család jól kielégitené a felhasználók különböző szintü igényeit és különböző gazdasági lehetőségeit. A megfogalmazott adatelőkészitő egység fokozza a fényszedő rendszerek teljesitményét és hatásfokát.

IRODALOMJEGYZÉK

[1] SCHMITT, GÜNTER: Fényszedés MK. 1983.

[2] KAS I. - PALOTÁSI A.: Lézerplotter gépkönyv, 1984.

[3] VARGA GYÖRGY: Egységes nyomdaipari szövegfeldolgozó rendszer létrehozásának lehetőségei számítástechnikai eszközökkel.
Doktori disszertáció, BME 1982.

[4] KNUTH, D.E. PLASS, M.F.: Bracking paragraphs into lines software practice and experience 1981.

[5] KNUTH, D.E.: TEX, 1981.

[6] REID, B.K.: A high level approach to computer document formating.
Conf. 7th: Annual ACM. Symp. on Principles of Programming Languages, Las Vegas, 1980.

[7] DÉVAI FERENC: Interaktiv szövegszerkesztés.
MTA SZTAKI, Working Paper, 1983.

[8] BYCKLING, EERO: Data flow in digital page production systems.
Report TKK-F-A518
Helsinki University of Technology, 1983.

[9] LASERCOMP Ismertető, 1983.

[10] GARA MIKLÓS: Nyomdai kisenciklopédia, MK. 1979.

LG-1 NYOMDAI ADATELŐKÉSZÍTŐ EGYSÉG

ESZENSZKI J., HÉVIZI L., dr. KAS I., PALOTÁSI A., Dr. VÖRÖS K.

1. BEVEZETÉS

A tanulmány tárgya egy olyan berendezés, amely lehetővé teszi az MTA SZTAKI-ban kifejlesztett LG1 tipusu lézerplotter csatlakoztatását korszerü fényszedő rendszerekhez.

A korszerü fényszedési technológia a kézirat rögzitésétől a nyomtató eredeti előállitásáig felöleli a nyomdai müveleteket. Számitógépes szövegfeldolgozó egységet alkalmaznak a szöveg nyomdai szedéséhez és számitógéppel vezérelt lézersugár állitja elő a fényérzékeny rögzitőn a nyomtatvány másoló eredetijét. Erre a célra a mai technológiák elsősorban fényérzékeny filmet használnak.

A nyomdai adatelőkészitő egység feladata az, hogy a számitógépes szövegfeldolgozó egységgel szerkesztett szöveget alkalmassá tegye az LG1 lézerplotterrel való levilágitásra. A tanulmány ezen berendezés lehetséges kialakitásával és müködésével foglalkozik.

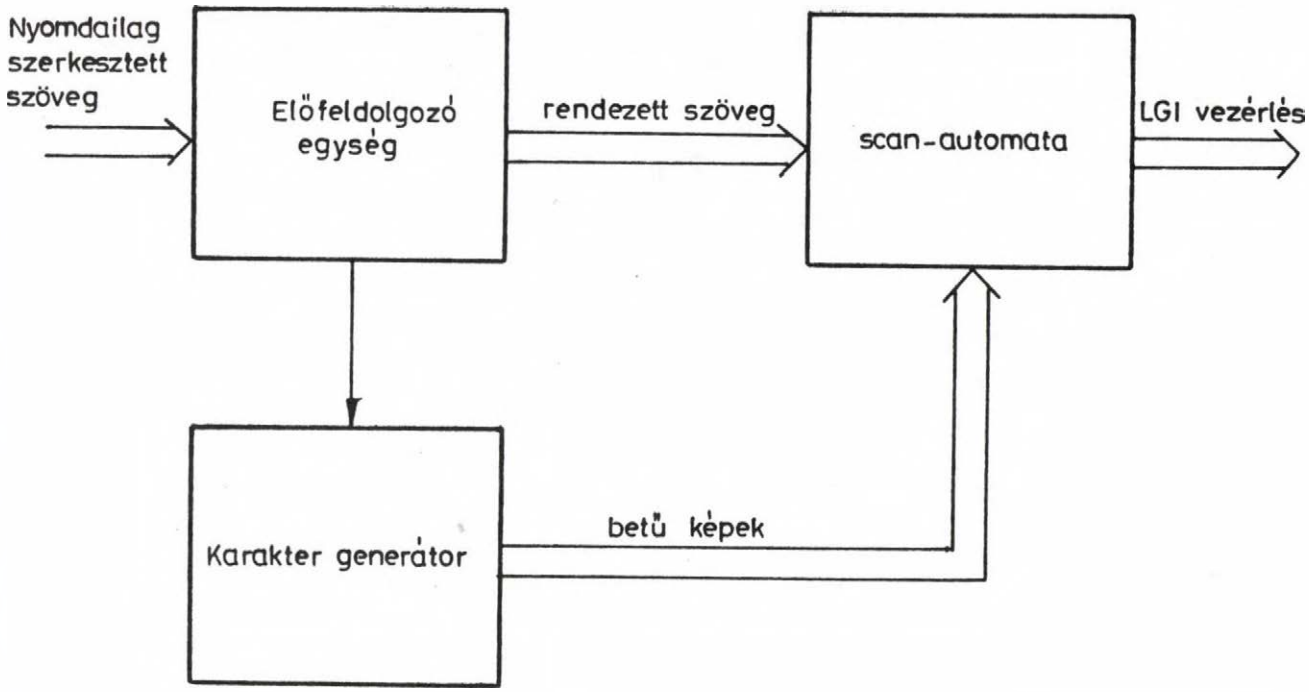
2. A NYOMDAI ADATELŐKÉSZÍTŐ EGYSÉG FELADATA

Az LG1 lézerplotterhez illeszkedő nyomdai adatelőkészitő egység a nyomdailag megszerkesztett szöveget oldalakra tördelten kapja. A levilágitandó oldalakat méretük szerint elhelyezi az exponálási felületen ugy, hogy az oldalakon a vizszintes sorok iránya az LG1 scanelési irányával egybeesik. A nyomdailag megszerkesztett szöveg a nyomtatandó karakterek kódjain kivül tartalmazza a betük tipusára, méretére vonatkozó utasitásokat, valamint közvetett és közvetlen pozicionáló utasitásokat. A közvetett utasitásra példa a sorvég és a szóköz jel, mig a közvetlenre a koordináta megadás. Az előzőek lebonthatók az utóbbiakra. A nyomdailag megszerkesztett szöveg tartalmazhat még levilágitást vezérlő parancsokat is. Például inverz módban fehér betükkel fekete háttérre ir a levilágitó. Az adatelőkészitő egység és a levilágitó nem végez semmiféle szerkesztői müveletet, igy az átvett szöveg nem tartalmazhat például kizárásra vonatkozó utasitást.

A nyomdai adatelőkészitő egység feladata az, hogy a fentiek szerint szerkesztett szöveg alapján 25 µm-es raszterpontokkal előállitsa az LG1 lézerplotter számára a levilágitás vezérlését a levilágitás ütemének megfelelő sebességgel. Az utóbbi feltételt nem szükséges kielégiteni olyan esetekben, amikor kezelői beavatkozás vagy betütipusok váltása szükséges a levilágitáshoz.

Az adatelőkészitő egység három fő részből áll. /1. ábra/

1 Előfeldolgozó egység
2 Karaktergenerátor
3 Scan-automata.



1. ábra
Adatelőkészitő egység blokk sémája

2.1. ELŐFELDOLGOZÓ EGYSÉG

Az előfeldolgozó egység adathordozón kapja a nyomdai szerkesztő számitógéptől a levilágitandó szöveget. A szöveg megadása a következőképpen történik:

A betük kódjai olvasási sorrendben érkeznek. A betütipus és betüméret váltást utasitás jelzi a megfelelő helyen. Van sorvég és oldalvég jel. Pozicionáló utasitás akkor kerül az adathalmazba, ha a következő megadott betü nem illeszkedik az előző betühöz. A betüközt és sorközt külön utasitással, vagy pozicionálással lehet előállitani. A későbbiekben részletesen tárgyaljuk a lehetséges szedési utasitásokat.

Az oldalak ugy vannak elhelyezve a filmen, hogy a scanek az oldalakat vizszintes irányban szelik át, és azokon a levilágitás felülről lefelé haladva történik. Tehát a lézerplotter a levilágitáshoz az egyes betüket a függőleges előfordulásuk sorrendjében igényli. Ez a sorrend nem egyezik a szerkesztés eredményeként létrejövő olvasási sorrenddel.

Az előfeldolgozó egység beolvassa az egyszeri levilágitás teljes adatállományát, majd az adatokat ugy rendezi, hogy a levilágitandó minták függőleges előfordulásuk szerint sorakozzanak és pozicionálásuk ne az oldalon belül, hanem a teljes levilágitási felületre, a 25 µm-es raszteregységekben történjék. Az ilymódon rendezett adathalmazt a levilágitás ütemében adagolva adja át a scan-automatának.

Az előfeldolgozó egység kezdeményezi a szövegben előforduló betük előállitását. A rendezés során kigyüjti a szövegben használt betütipusokat, betüméreteket és továbbitja a karaktergenerátornak.

2.2. KARAKTERGENERÁTOR

A levilágitáshoz közvetlenül a betük bittérképére van szükség. A 25 µm-es felbontásu bittérképek tárolása már kis számu betü esetén is nagy feladat, és sokféle betütipus különféle méretü változatainál a tárolás megoldhatatlan. Ezért célszerü háttértárakon a betük alakját közvetlenül jellemző pontokat, vagy függvényeket rögziteni, amelyek helyigénye még elfogadható. Ezen adatokból a felhasználás előtt kell az adott méretü betük tömöritett bittérké-

pét előállitani, amelyeket a továbbiakban a felhasználással egyidőben kell kifejteni teljes bittérképpé.

A karaktergenerátor az előfeldolgozó egységtől átveszi a levilágitásra kerülő betüket azonositó kódokat, majd előállitja a betük tömöritett bittérképét és eltárolja egy olyan memóriában, amelyhez a scan-automata is hozzáférhet.

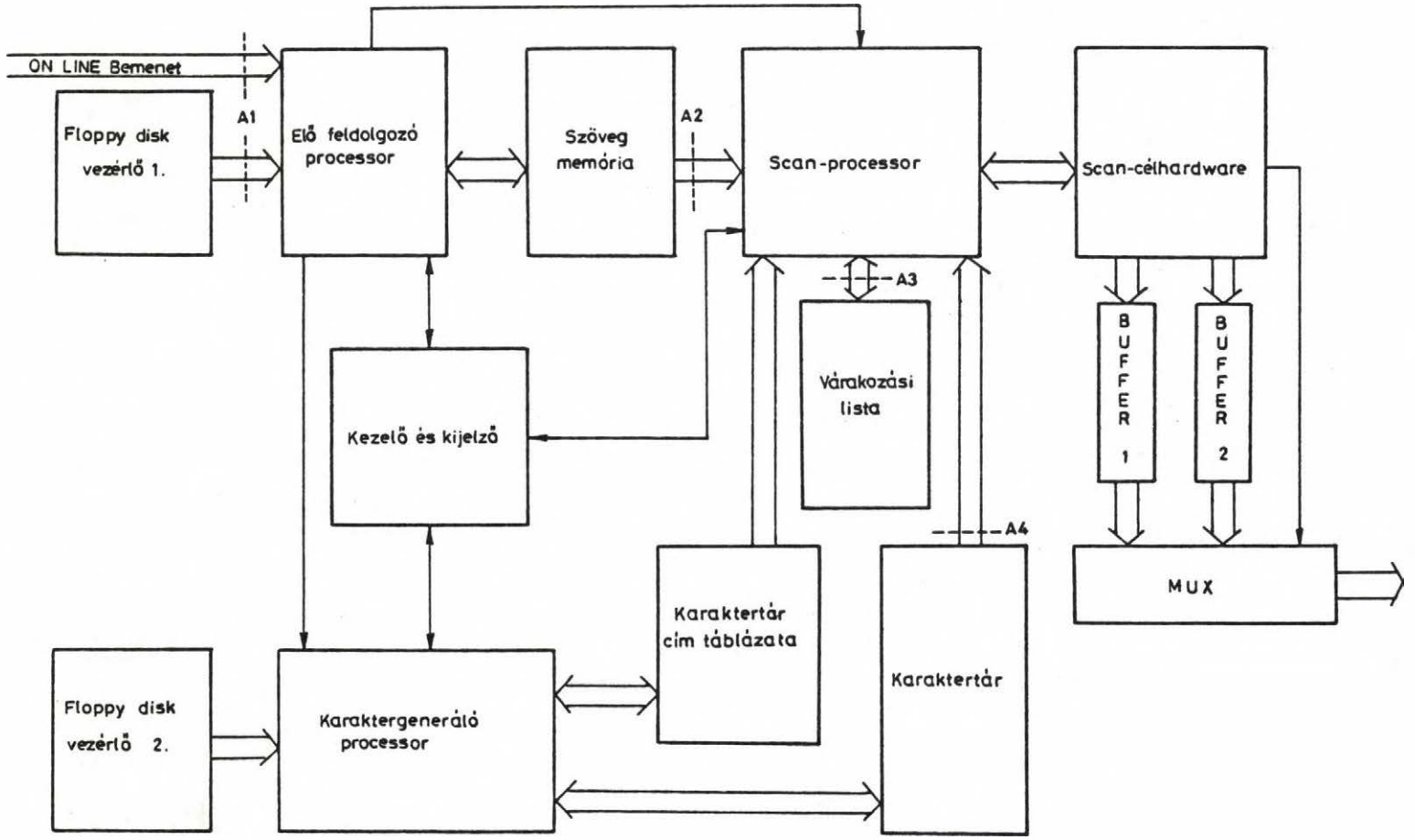
2.3. SCAN-AUTOMATA

Az adatelőkészitő egység közvetlen vonalon vezérli a lézerplottert, amelynek a tervezett fényszedési teljesitmény eléréséhez a lehetőségek szerint teljes sebességgel, megszakitás nélkül kell müködnie. A lézerplotter bemeneti adatait is ugyanevvel a sebességgel kell előállitani, és az előállitásban a scan-automatának legalább egy lépéssel, vagyis egy scannel a levilágitás előtt kell járnia.
Az automata az előfeldolgozó egységtől átveszi a levilágitandó betük kódjait, amikor azok a levilágitáskor sorra kerülnek, és mindaddig emlékezik rájuk, mig a betük levilágitása be nem fejeződik. A scaneket a betük vizszintes irányu szeletei alkotják, amelyeket a scan-automata a karaktergenerátor által feltöltött memóriában talál meg. A scan-automata scanenként váltva, két bufferben dolgozik. Mig az egyik bufferből a levilágitást vezérli, addig a másikban elkésziti a következő scant.

3. AZ ADATELŐKÉSZÍTŐ EGYSÉG FELÉPÍTÉSE

Az adatelőkészitő egység három jól elkülönithető része közül az előfeldolgozó egység teljesen, a karaktergenerátor jórészt önállóan dolgozhat, mig a scan-automata az előző két egység "termékeit" hasznositja, és feladatának tekintélyes része adatmozgatás. Az adatelőkészitő müködésének első fázisában, közvetlenül a szöveg beolvasása után az előfeldolgozó egység és a karaktergenerátor dolgozik. A scan-automata csak egy bizonyos idő elteltével kezdi adatokkal táplálni a lézerplottert. A levilágitás ideje alatt a karaktergenerálás folytatódik, ha olyan betüre van szükség, amely nem érhető el tömöritett bittérkép formájában.

A 2. ábra az adatelőkészitő egység felépitését mutatja. A funkcionális egységek müködését és a megjelölt csatlakozási felületeken áramló adatokat a következőkben ismertetjük.



2. ábra

Az adatelőkészitő egység részletes blokk sémája

3.1. FLOPPY DISZKVEZÉRLŐ 1

Ezen az adatbeviteli csatornán, vagy az ON-LINE vonalon kerül az adatelőkészitő egységbe a nyomdailag megszerkesztett szöveg.

3.2. ELŐFELDOLGOZÓ PROCESSZOR

Az előfeldolgozó processzor egyszeri levilágitáshoz elegendő kiszedett szöveget olvas be és helyez el a szövegmemóriában. Ezt a szöveget ugy rendezi, hogy az egyes betük a levilágitás sorrendjében legyenek egymás után és a betük megadásán kivül már csak egyszerü koordináta utasitások maradjanak az adathalmazban. A bemenő nyelvet az A1, mig az utóbbi, rendezés utáni adatstrukturát az A2 csatlakozási felület kapcsán tárgyaljuk.

Az előfeldolgozó processzor külön vonalon, a levilágitandó betük kódjainak elküldésével adja a karaktergeneráló processzor tudtára, hogy mely betük tömöritett bittérképét állitsa elő és töltse be a betütárba.

3.3. SZÖVEGMEMÓRIA

Olyan kétportos memória, amely irható-olvasható az előfeldolgozó processzor és csak olvasható a scan-processzor számára. Az előfeldolgozási müvelet végére tartalma átalakul, de továbbra is az egyszeri alkalommal levilágitandó szöveg egyértelmü, de már a levilágitási sorrendhez alkalmazkodó leirását tartalmazza.

3.4. FLOPPY DISZKVEZÉRLŐ 2

Ez a floppy diszk vezérlő olvassa be azokat az adatokat, amelyeket a karaktergeneráló processzor a betük tömöritett bittérképeinek előállitásához felhasznál. A diszken az egyes betütipusok betükészlete a betümérettől független módszerrel, lényegesen kevesebb adattal van leirva, mintha tömöritett bittérképpel lenne.

3.5. KARAKTERGENERÁLÓ PROCESSZOR

A karaktergeneráló processzor az előfeldolgozó processzortól megkapja azon betük kódjait, amely betüket tömöritett bittérképes formában elő kell készitení a levilágitáshoz. A tömöritett bittérképek előállitása bonyolult számitási müveletekkel jár, amelyek időigényesek, igy a gyakori betütipus és betüméret váltások a folyamatos levilágitását hátráltathatják.

A betük tömöritett bittérképei a karaktertárba kerülnek, ahol gyorsan elérhetők a scan-processzor számára. A karaktertár pillanatnyi tartalmáról a karaktertár cimtáblázata ad felvilágositást, amelyet szintén a karaktergeneráló processzor kezel.

3.6. KARAKTERTÁR

Néhány átlagos méretü betükészlet tömöritett bittérképeit képes egyidejüleg tárolni. A karaktergenerátor számára irható-olvasható, a scan-processzor számára olvasható kétportu memória. A tárban lévő tömöritett bittérképek adatstrukturáját az A4 csatlakozási felülettel kapcsolatban fogjuk részletezni.

3.7. KARAKTERTÁR CIMTÁBLÁZATA

A karaktertárban elhelyezett betük tömöritett bittérképes leirásának kezdőcimét és a betük azonositó kódját tartalmazza. A karaktertár adatainak gyors elérhetőségét szolgálja. Hasonlóan a karaktertárhoz, ez a memória is kétportu, irható-olvasható a karaktergeneráló processzor és olvasható a scan-processzor számára.

3.8. SCAN-PROCESSZOR

A scan-képek összeállitását vezérli. A levilágitásnak megfelelő ütemben beolvassa a szövegmemóriából az egymást követő betük kódjait, majd a karaktertár cimtáblázatából kikeresi a betük tömöritett bittérképének karaktertárbeli kezdőcimeit. A scan összeállitásában olymódon vesz részt, hogy a karaktertárban megkeresi az éppen beépitendő betü megfelelő szeletét, és az adatokkal a scan-célhardware-t táplálja. Azon betüket, amelyeket az éppen összeállitás alatt álló scan metsz, a várakozási listáról azonositja. A várakozási lista tartalmát scanről-scanre haladva korszerüsiti. A várakozási listán mindig követi a betük scanbe-fejtésének menetét.

3.9. VÁRAKOZÁSI LISTA

A várakozási lista olyan memóriaterület, amelyet a scan-processzor irhat, olvashat. Az éppen scanbe fejtés alatt álló betükről a következő információkat tárolja:

> Hol helyezkedik el a betü scanen belül?
> A betü mely szeleténél tart a scan-összeállitás, illetve milyen memóriacimről kell kivenni a betü tömöritett bittérképének következő szeletét.

A várakozási listán levő adatok szerkezetét az A3 csatlakozási felület kapcsán részletezzük.

3.10. SCAN-CÉLHARDWARE

A scan-célhardware a scan-processzor vezérlése alatt a betük tömöritett bittérképeit scanbe-fejti. A scan-képeknek az összeállitás során memória buffer felel meg.

3.11. BUFFER 1, BUFFER 2 ÉS MUX

A váltóbufferes megoldás azt teszi lehetővé, hogy mig egy scan-kép elkészül az egyik bufferben, addig a másikból történjen az LG1 lézerplotter vezérlése. A váltást a scan--célhardware a MUX multiplexer révén végzi.

3.12. KIJELZŐ ÉS KEZELŐ EGYSÉG

A berendezés mindhárom fő egységével kapcsolatban áll. A kezelő személy a berendezés müködését a kijelzőn figyelemmel kisérheti és a müködésbe a kezelőpulton beavatkozhat.

4. AZ ADATELŐKÉSZÍTŐ EGYSÉGGEL SZEMBEN TÁMASZTOTT KÖVETELMÉNYEK

A berendezés leirását az LG1 lézerplotter felöli oldalról kezdjük, mivel ezen az oldalon a kapcsolódási interface már kialakult és rögzitett. A lézerplotter bemeneti specifikációjából és müködési sebességéből kiindulva egyértelmü következtetéseket fogalmazhatunk meg az adatelőkészitő egység felépitésére vonatkozóan.

4.1. A LÉZERPLOTTER RAJZMÉRETE 480 x 540 mm ÉS A SCANELÉS AZ 540 mm-ES HOSSZANTI IRÁNYBAN TÖRTÉNIK

Ezt figyelembevéve durva becslés adható a lehető legnagyobb adatmennyiségre, amelyre egyszeri levilágitáskor szükség lehet. Ezt nyomdailag megszerkesztett formában a szövegmemóriában kell tárolni. Teljes kitöltést feltételezv a legkisebb gyémánt betüből körülbelül 170 000 db, mig a garmond betüből 55 000 db fér el a rajzfelületen. Mivel a valóságban sohasem fordul elő a kisbetük ilyen sürü nyomtatása, a szövegmemória méretét 128 kByte-ra választani elegendő.

4.2. A LÉZERPLOTTER FELBONTÁSA 25 µm, MINDKÉT IRÁNYBAN, A SCAN-SZÉLESSÉG 200 µm, AZAZ 8 RASZTERPONT ÉS A SCAN--HOSSZUSÁG 21400 RASZTERPONT

A felbontás finomságából következik, hogy a tervezett 0.1 mm-es nyomdatechnikai pozicionálást az adatelőkészitő egység könnyen kezelheti. Az egyetlen gond, hogy amig a lézerplotter bemeneti adatstrukturája a 8 raszterpontos, azaz scan-szélességnyi tömöritésen alapul, addig a nyomdai alkalmazásban a 0.1 mm-es félscannenkénti tömörités látszik előnyösnek. Ezt a gondot a scan-célhardware fogja megoldani. Tehát a buffer 1 és 2-ben a bittérképek 8 bit, a betütárban pedig 4 bit szélességre vannak tömöritve.

A teljes rajzfelületen való pozicionáláshoz 13 vagy 15 bit szükséges, attól függően, hogy 0.1 mm vagy 25 µm-es raszter az egység.

4.3. A LÉZERPLOTTER 8 PERC ALATT VILÁGITJA LE A TELJES RAJZFELÜLETET, AZAZ 1 mm-T HALAD 1 MÁSODPERC ALATT

Ez a levilágitási sebesség azt követeli meg az adat-előkészitő egységtől, hogy 200 ms-onként előállitson egy scan-képet. A scan-kép kiteritett bittérkép formájában 21400 byte hosszu. 20 %-os tömöritést feltételezve leröviditheto 4280 byte-ra. Ez azt jelenti, hogy 1 byte adatot átlagosan 40 µs alatt kell előállitani és elhelyezni a bufferek valamelyikében.

A bufferekben az adatok már az LG1 bemenő nyelvén tárolódnak.

5. GÉPEN BELÜLI ADATSTRUKTÚRÁK

5.1. BEMENŐ NYELV AZ A1 CSATLAKOZÁSI FELÜLETEN

A szerkesztő-számitógép a szöveget a következő formában adja át:

- Az adathalmazt egy fejléc előzi meg, amely alapján a szöveg azonositható.

- A betük alapvonalra illeszkednek, és kódjaik az alapvonalon balról jobbra haladva következnek a leirásban. /1.3.a. ábra/.

- Az egymást követő betük pozicionálási utasitás hiányában közös alapvonalra, befoglaló formáikkal érintkezve egymás mellé lesznek elhelyezve. Az aktuális vizszintes koordináta minden betü beolvasásával módosul.

- A pozicionáló utasitások az oldalon belül érvényesek, és a viszonyitási pont az oldal bal fölsõ sarka.

- Az alapvonal helye függõleges pozicionáló utasitás hatására elmozdul.

- Amennyiben egy betü nem záródik az elõtte levõhöz, a betü bal szélét vizszintes pozicionáló, vagy szóköz utasitással lehet eltolni az alapvonalon.

- A szöveg oldalakra tördelve érkezik. Minden oldal a lapméretek és az első sor alapvonalának megadásával kezdõdik, és oldalvég utasitással végzõdik. A rajzfelületen az adatelõkészitõ egység helyezi el az oldalakat.

- Téglalapok rajzolása lehetséges. Méreteiket utasitással lehet megadni, oldalaik a lapszélekkel párhuzamosak.

A téglalap pozicionálása a bal felső sarka szerint történik.

- További rajzolási módot meghatározó utasitások definiálhatók. Ilyen az inverz mód, vagy a sorvég és a sorköz utasitás.

- A pozicionáló és méret utasitások egységként 0.1 mm-t vagy 25 µm-es rasztertávolságot használhatnak.

5.2. AZ ELÕFELDOLGOZÁS SORÁN LÉTREJÖVÕ ADATHALMAZ AZ A2 CSATLAKOZÁSI FELÜLETEN

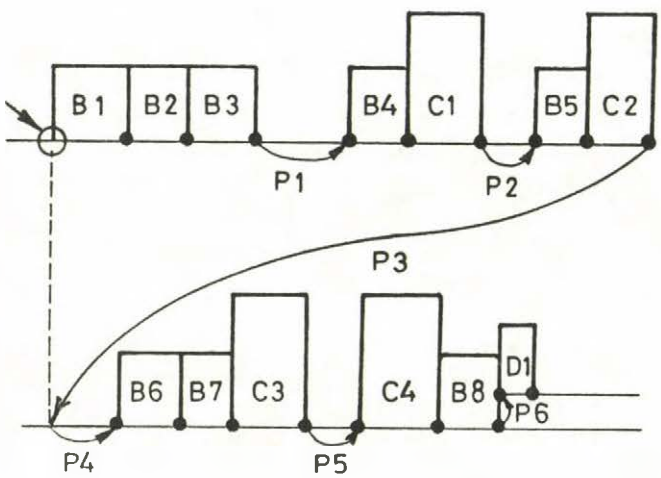
Az elõfeldolgozás eredményeként a szövegmemóriában a teljes rajzméret szöveganyaga tárolódik a levilágitási sorrendben.

- Teljes ramzméretre történő rendezéssel a levilá-gitandó betük és téglalapok a bal felső sarkuk poziciója szerint felülről lefelé mutató irányban sorba vannak állitva, azaz annak a betünek a kódja szerepel előbb a szövegmemóriában, amelynek bal felső sarka közelebb van a levilágitási felület felső határvonalához. /3.b. ábra/

- A betük kódjai között bármely helyen előfordulhat betütipust és betüméretet meghatározó utasitás, de itt már más elrendezést jelent mint a bemenő nyelvben. A 3. ábrán példa szemlélteti a különbséget.

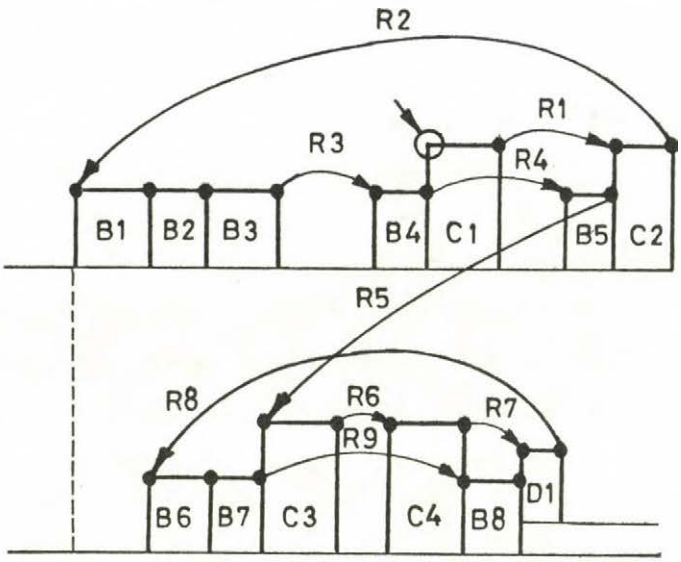
- A pozicionáló utasitások a teljes rajzfelületre vonatkoznak és nem fordulhat elő függőleges irányu vissza-pozicionálás. A viszonyitási pont a rajzfelület bal fölső sarka. A betükód automatikusan a betüszélességgel növeli az aktuális vizszintes koordinátát.

- Más utasitás nem megengedett.



3.a. ábra

A betük elrendeződése a bemenő nyelvben



3.b. ábra

A betük az irányitott rendezés után

B1 - B8 C1 - C4 D1	egy adott betütipus adott méretü betüinek befoglaló formáit jelképezi és kódjait jelenti.

A fenti szöveg leirása a bemenő nylevben /A1/ a következő sorrendben van:

Kiindulási pozició az ábrán üres karikával és nyillal jelölt pont. P1, P2, P4, P5, P6 pozicionáló utasitásokat és P3 "uj sor" utasitást jelöl. /3.a. ábra/

..., B1,B2,B3,P1,B4, betüméret váltás C-re, C1, P2, betüméret váltás B-re, B5, betüméret váltás C-re, C2,P3,P4, betüméret váltás B-re, B6, B7, betüméret váltás C-re, C3, P5, C4, betüméret váltás B-re, B8, P6, betüméret váltás D-re, D1,...

A fenti szöveg rendezés után /A2/

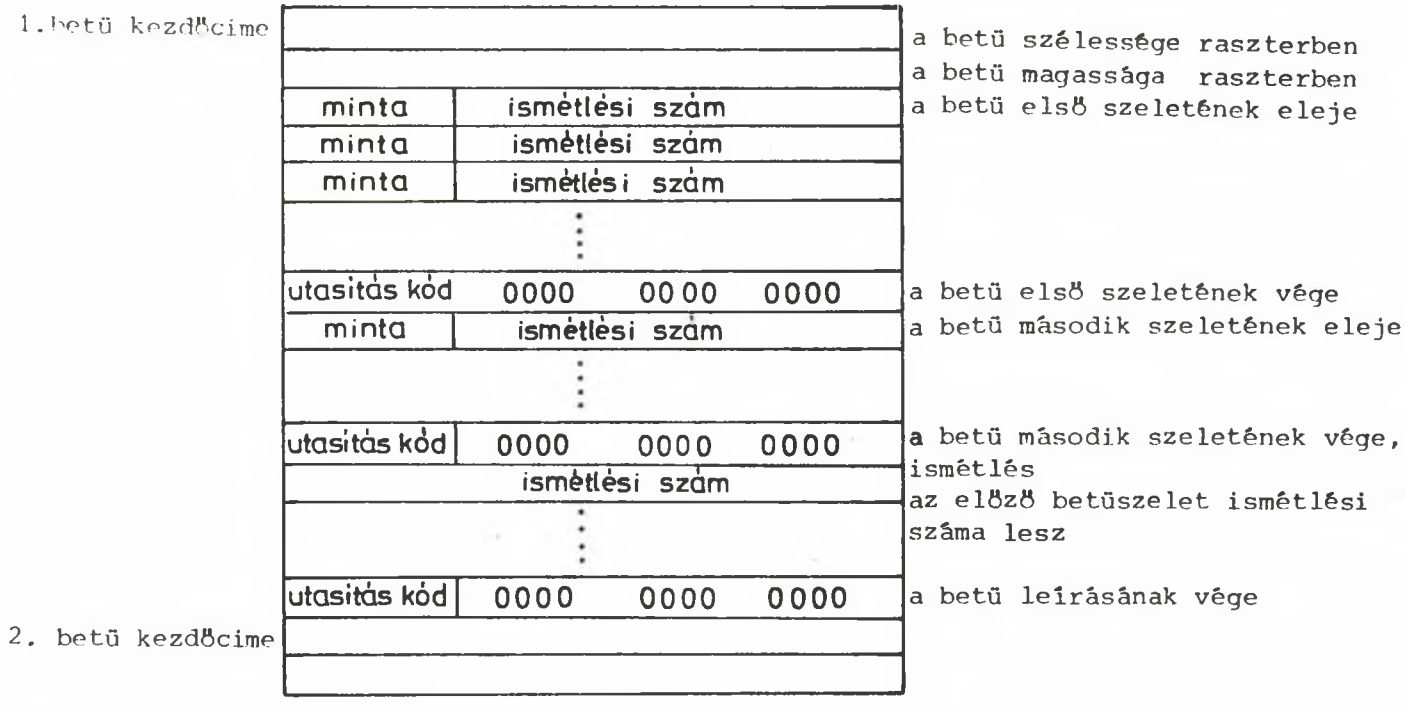
Kiindulási pozició a tömör karikával és nyillal jelölt pont, és R1, R2, R3, R4, R5, R6, R7, R8, R9 pozicionáló utasitások. /3.b. ábra/.

..., C1,R1,C2,R2, betüméret váltás B-re, B1,B2,B3,R3,B4, R4,B5,R5, betüméret váltás C-re, C3, R6, C4, R7, betüméret váltás P-re, P1, R8, betüméret váltás B-re, B6, B7, R9, B8,...

5.3. A BETÜK TÖMÖRITETT BITTÉRKÉPÉNEK TÁROLÁSA A KARAKTERTÁRBAN, AZ A4 CSATLAKOZÁSI FELÜLETEN

A karaktertér célszerüen 16 bites szóhosszuságra szervezett. Az egyes betük tömöritett bittérképei folytatólagosan helyezkednek el benne. A karaktertár cimtáblázata tartalmazza a karaktertárban szereplő betüket meghatározó adatokat /betütipus, betüméret, betükód/ és hozzájuk tartozó tömöritett bittérképek karaktertárbeli kezdőcimét. A cimtáblázatban a betüt meghatározó adatok szerint lehet megkeresni a kezdőcimet.

A betük tömöritése 0.1 mm-széles szeletekben /fél scanre, 4 bitre/ történik. A 16 bitből az első 4 adja a mintát és a maradék 12 az ismétlési számot. Tehát a legnagyobb ismétlési szám 4095 lehet, ami egy mintának max. 102 mm hosszuságu csikját képes leirni. Az ismétlési szám 8 bites szóhossz esetén legfeljebb 15 lehetne, ami 0.375 mm-nek felel meg. Ez az ismétlési hossz a legtöbb esetben nem elegendő, igy be kellene vezetni a kétbyte-os utasitásokat is. A zérus ismétlési számnak nincs értelme, igy ezt célszerü felhasználni olyan utasitások céljára, mint például "betüszelet vége", "betüszelet ismétlődési szám a következő szó tartalma", vagy a "betü tömöritett bittérképes leirásának vége". Az egyes utasitásokat ilyenkor a minta helyén levő 4 bit kódolhatja. Összefoglalva a karaktertárban lévő adatok a 4. ábra szerint helyezkednek el:



4. ábra

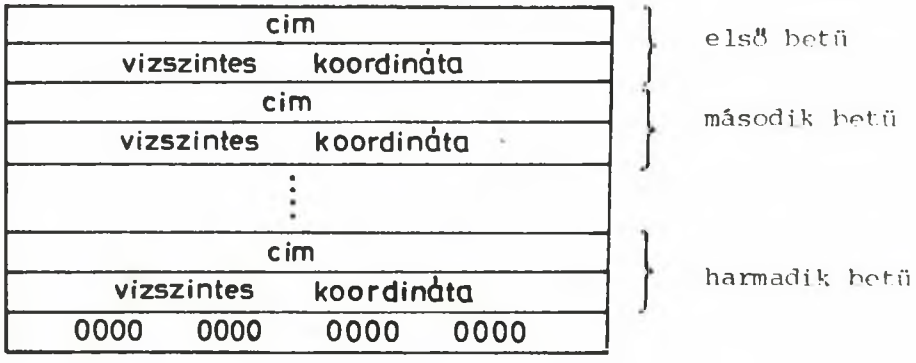
Adat struktura a karaktertárban

A karaktertárat ugy kell kialakitani, hogy gyorsan olvasható-irható legyen, és képes legyen tárolni 2-3 betütipus teljes betükészletét a leggyakrabban előforduló közepes méretben.

5.4. A VÁRAKOZÁSI LISTA ÉS AZ A3 CSATLAKOZÁSI FELÜLET

A várakozási lista egy táblázat, amely tartalmazza azon betük és téglalapok szükséges adatait, amelyeket az éppen összeállitás alatt álló scan átszel. Amikor egy scant elkészit a scan-processzor, kiolvassa a szövegmemóriából a következő utasitást vagy betükódot. Ha szükséges uj aktuális poziciót számol. Ha csak utasitásokat talál, mindaddig ismétli a beolvasást, amig betükódra nem lel.

Ha betükódot talált és az aktuális pozició a scanbe esik, akkor a betüt fel kell vennie a várakozási listára. Ehhez először a betü karaktertárbeli kezdőcimét kikeresi a karaktertár cimtáblázatából, majd kiemeli a karaktertárból a betüszélességét, amellyel megnöveli az aktuális vizszintes koordinátát. A korábbi aktuális vizszintes koordinátát, és a karaktertárbeli kezdőcim +2-t, a betü első szeletének kezdőcimét együtt elhelyezi a várakozási listán. /5. ábra/.



5. ábra

A várakozási lista adat strukturája

Amig a betük bal felsö sarka beleesik scanbe, addig minden betü felkerül a várakozási listára, tehát a scan-processzor a szövegmemóriából egymás után beolvassa a betükódokat és a fenti müveleteket ismétli. Ha az aktuális pozició kikerül a scanböl, akkor elkezdi a scan tényleges összeállitását. Végigszalad a várakozási lista elemein és minden elemmel kapcsolatban elvégzi az alábbiakat.

A vizszintes koordinátát rögtön átadja a scan-célhardware-nek, és utána sorra a tárolt cimtől kezdödően a karaktertárból a betüszelet adatait is. A szelet véget a 0 ismétlési számról és a mintáról felismeri. Ha az egyben a betütárbeli végét is jelöli, akkor törli a betüt a várakozási listáról, azaz 0-t ir a cim helyére, különben pedig a következö tömöritett betütérkép szelet kezdöcimét tárolja el a régi helyett. Igy nem szükséges külön megjegyeznie, hogy hány szeletet rajzolt már le a betüböl.

Befeketitett téglalap rajzolása hasonlóan történik, mint a betüké. Elözöleg a karaktergeneráló processzor a téglalap méretének megfelelően elkésziti és elhelyezi a karaktertárban annak tömöritett bittérképét.

A várakozási lista méretét célszerü 2 kszó-ra választani, mert a legkisebb betüknek a lehető legtömörebb elhelyezésénél sem lehetséges, hogy 1000 betünél több essen egy scan-re. Az 1000 betü igy is két egymással sorköz nélkül érintkező sorban helyezkedne el.

6. AZ ADATELŐKÉSZÍTŐ EGYSÉG MŰKÖDÉSÉNEK ÖSSZEFOGLALÁSA

A berendezés floppy-diszkről beolvassa az egyszeri levilágitáshoz elkészitett nyomdailag megszerkesztett szöveget, és tárolja a szövegmemóriában. A szövegen előfeldolgozást végez, amely során a szövegben található szedési utasitásokat értelmezi és egyszerübbekre bontja, majd a teljes betüanyagot a tényleges térbeli elhelyezésük szerint rendezi. Kigyüjti a szövegben előforduló betütipusokat és méreteket és elinditja azok tömöritett bittérképeinek előállitását. A berendezésben ezt a munkát külön karaktergenerátor végzi, amely floppy-diszken tárolt absztrakt betüleirásokat használ forrás-adatként. A betük tömöritett bittérképei karaktertárba kerülnek és gyorsan elérhetők, amikor a betük egy-egy szeletének scanbe-fejtése történik. Ez utóbbi müveletet nagysebességü célhardware végzi egy külön processzor irányitása alatt. Ez a processzor a rendezett betühalmazból mindig annyit kezel, amennyire az egyes scanek összeállitásához szükség van, és folyamatosan táplálja adatokkal a célhardware-t. A célhardware váltóbufferbe tölti az LG1 bemenő nyelvü adatokat scanenként. Mialatt az egyik bufferben az összeállitás, addig a másikból a levilágitás történik.

7. A BERENDEZÉS HARDWARE KIALAKÍTÁSÁRÓL

Az adatelőkészitő három részegysége egy-egy mikroprocesszoros környezetet jelent. Az előfeldolgozó egység a szerkesztő-számitógéptől átvett 8 bites szervezésü adatokon dolgozik. Az előfeldolgozásnál a végrehajtási sebesség nem döntő fontosságu, igy erre a feladatra egy 8 bites rendszer alkalmas, amely rendelkezik floppy-diszk illesztővel, két-

irányu vonallal a kijelző-kezelő egység, a karaktergenerátor és a scan-processzor felé. Az előfeldolgozó processzor külön programmemóriához és a szövegmemóriához férhet hozzá. Az utóbbi kétportu memória, amelyet a scan-processzor is olvashat.

Mivel a scan-processzor elsősorban a tömöritett bittérképeket és a koordinátákat használja adatként, amelyek 16 bitesek, célszerü 16 bitesre választani magát a mikroprocesszort is. A scan-processzor soros vonalon kapcsolatot tart a másik két mikroprocesszorral, a kijelző-kezelő egységgel és párhuzamos vonalon a scan-célhardware-rel. Jól használható a 16 bites mikroprocesszor nagy cimtartománya a nagyméretü memóriák kezelésekor. A scan-processzornak gyors adatátviteleket és kereséseket kell elvégeznie a karaktertár-ban, a szövegmemóriában, a karaktertár cimtáblázatában és a várakozási listában. Ez a processzor központi szerepet kap a belső adatforgalomban és a teljes berendezés müködését ütemezi a levilágitási sebességgel összhangban.

A helyzet más a karaktergeneráló-processzor esetében. A legfontosabb követelmény vele szemben az, hogy a lehető leggyorsabban állitsa elő a bekért betük tömöritett bittérképét, és a karaktertár cimtáblázatában pontosan kövesse a karaktertár tartalmát. A karaktergenerátor müködési ideje nagyrészt holtidő lesz a levilágitásban. A 16-bites adatstruktura és a sebességi követelmények miatt célszerü ezt a processzort is 16 bitesre választani. A karaktergenerátorhoz ugyanolyan floppy-diszk vezérlő kapcsolódik, mint az előfeldolgozó processzorhoz.

NYOMDAI BETŰK ELŐKÉSZÍTÉSE FÉNYSZEDÉSRE

dr. LAUFER J., SZŐNYI T.

1. BEVEZETÉS

A ma használatos nyomdatechnikai eljárások mellett, a nyomdai termékek előállitása két, jól elkülönülő, de egymáshoz kapcsolódó folyamatból áll. Először a kefelenyomatot állitják elő, majd a kefelenyomatról azt a klisét, amivel a sorozattermék készül. A kefelenyomat előállitásának lépései, szedés, tördelés, és természetesen a javitások.

A kefelenyomat előállitására használt hagyományos eljárások a valószinüleg mindenki számára ismert Gutenberg féle nyomtatási eljárásban gyökereznek. Ennek lényege, hogy a betüket ólomból öntik ki oly módon, hogy kis téglalap alapu hasáb alapjából reliefszerüen domborodik ki a betü vagy jel rajzolata. Az emlitett kis ólomhasábot betübélyegnek nevezik. A betübélyegeket egy keretbe szorosan egymás mellé szedik és igy alakulnak ki a nyomtatandó szöveg szavai, illetve egymás alatt a sorok. Ott, ahol a papiron üres helyet kivánnak hagyni, megfelelő méretü, rajzolat nélküli ólomhasábot helyeznek a keretbe. Végül, ha kialakult az oldal, festéket kennek a fémbetükre, és a papirlapot a keretre simitva megkapják a nyomtatott lapot. Ha a nyomaton hibát találnak, a megfelelő betüket kicserélik a keretben, és máris előáll a javitott változat.

A mult század végén kezdődött Gutenberg eljárásának korszerüsitése. Először a nyomat sokszorositását reformálták meg, kialakultak a rota eljárások. Következő lépésként az ólombetük előállitását, szedését automatizálták. Végül a fényszedés bevezetésével megkezdődött az elszakadás az ólom betütől.

A számitógépes fényszedés célja, hogy az ólombetüs szedést teljesen kiszoritva, a nyomat nyomdai igényeknek megfelelő minőségü fotókópiáját állitsa elő, amely kiindulásként szolgál a további sokszorositások számára. Müködésének elengedhetetlen feltétele a speciális periféria, amely fényérzékeny papirra, raszter pontokból a nyomat fotókópiáját előállitja. A speciális periféria karakterkészlete az a betükészlet, más néven betü font, amelyet a tipográfus kiválasztott az adott nyomtatvány számára, természetesen speciális leirásban.

Munkacsoportunk, egy a számitógépes szedés fejlesztését megcélzó programban vett részt. A mi munkánk volt mindazon programok elkészitése, amelyek szükségesek ahhoz, hogy a betük rajzolatából a számitógépes szedés és megjelenités számára érthető adattömeg legyen. Ólombetüs hasonlattal élve, egy számitógépes vésnök eszközeit készitettük el.

2. A NYOMDAI BETÜKRŐL

Mondhatni, hogy az irás megjelenése óta a feliratok, oklevelek elkészitéséhez használt betük alakja folyton változott. A változásnak több oka volt, részint változtak azok az anyagok, amivel és amire irtak, részint a betü mint kép alkalmazkodott az aktuális müvészeti stilusirányzathoz. Eközben a betük nem lettek mindig olvashatóbbak és néha érdekes kisérlet tárgyai voltak, igy a felvilágosodás korában, amikor csak egyenesekből és körökből álltak az ábécék betüi. Nagy müvészek, mint Leonardo da Vinci, Albert Dürer is foglalkoztak betü tervezéssel, de mellettük a nyomtatás is kitermelte saját betümetsző müvészeit. A legnagyobbak Garamonde és Bodoni. Európaszerte ismert betümetsző volt Misztótfalusi Kis Miklós, aki a latin ábécén tul az első örmény betükészletet is megalkotta. [2]

A nyomdai betü szerzőijoggal védett grafikai alkotásnak minősül. A grafikus a karakterek megrajzolásával azoknak nem csak alakját, hanem egymáshoz képesti helyzetét is meghatározza azáltal, hogy minden karakterhez megtervezi a betübélyeg nagyságát, és a karakter elhelyezkedését a bélyegen.

Érdekes kérdés a különböző méretü fontok előállitása. A nyomdászhagyomány figyelembe veszi, a látásban rejlő nemlinearitásokat. A cél nem az, hogy a különböző méretü karakterek részleteinek méretaránya állandó legyen, hanem az, hogy az emberi szem ezeket az arányokat állandónak érzékelje.

2.1. A NYOMDAI BETÜ MÉRETE

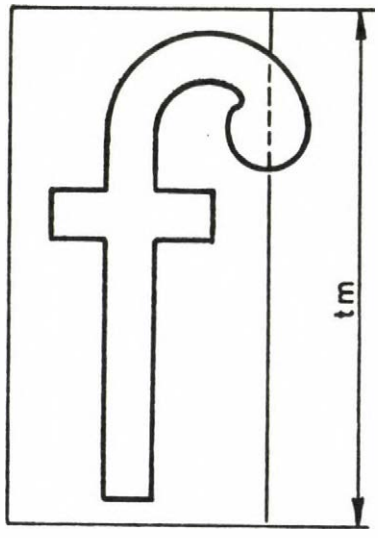
A nyomdai betük nagyságát ugynevezett Didot pontban mérik. /1 Didot pont = 0.376 mm/ A használatos mérettartományt három részre osztják. A tervező jelzi, hogy az általa készitett rajzolat, milyen nyomdai méret 50-szeres nagyitása. Az ebből a betüalapból lineáris kicsinyitéssel megengedett legkisebb méret az eredeti méret fele, legnagyobb pedig az eredeti méret kétszerese. A nyomdászhagyomány szerint a fenti mérethatárokon kivül eső méretek számára a tervezőnek uj

rajzolatot kell készitenie.

Érdekes megnézni, hogy mit ért a nyomda a betük méretén. A betük méretén hagyományosan a betübélyeg 1. ábrán jelzett méretét értik. Ennek a méretnek nincs egyértelmü kapcsolata egy-egy karakter tényleges méretével, inkább a bélyeg szedéskori megfogásával van kapcsolatban. Ugyanakkor a betü szélei "lelóghatnak" a bélyegről.

A fényszedés kialakitott egy uj méretrendszert: a font H-betüjének mm-ben mért magasságát nézik. Ennek a méretnek már szoros köze van a nyomat nyujtotta látványhoz, de semmi köze ahhoz a teljes távolsághoz, amit a betü a sorban elfoglalhat, /Az **Á** ékezetének tetejétől a g aljáig mért távolság/ jóllehet ez egy olyan méret, ami a fontból kialakitható sorok legkisebb sortávolságát meghatározza.

méret D.idot pontban =
=tm/0.376



1. ábra

A betü mérete

A hagyományos nyomdatechnika a karakterek rajzolatait 1000 vonal/inch finomsággal adja vissza. Ennek megfelelően, a számitógépes fényszedés 40 pont/mm-es osztással dolgozik. Más szavakkal, 25µ x 25µ -os raszterpontokból álló "bittérképen" ábrázolja a betüket.

Az alábbiakban a grafikus által rajzolt karakter feldolgozásával kapcsolatos problémákat és néhány lehetséges megoldást fogunk ismertetni.

3. A FELDOLGOZÁS LÉPÉSEI

3.1. A DIGITALIZÁLT KÉP

A rendszerünkben karakterek raszterpontokban való ábrázolására két elnevezést használunk. Digitalizált képnek nevezzük az eredeti grafika digitalizáló berendezéssel való letapogatásából nyert képet. Bittérképnek nevezzük a feldolgozás egy későbbi fázisában, program által előállitott raszterképet.

Az első kérdés, hogy mekkora felbontással digitalizáljunk egy képet, hisz minél nagyobb a felbontás, annál kisebb az információ vesztés. A körülményeket figyelembe véve, azt kivánjuk meg, hogy egy rajzolat legalább annyi pontra digitalizálódjék, mint ahány ponttal a feldolgozás végén, a nyomdai igényt figyelembe véve meg kell jeleniteni az adott méretben. A digitalizálóból nyert raszterkép elvben alkalmas lenne közvetlen levilágitásra, igy azonban az eredeti rajzolatnak valamilyen torzult változatát kapnánk vissza. A torzulás egyik oka, maga a digitalizálás. Néhány, a digitalizálásból eredő torzulás: sávok, legfeljebb két raszterponttal megszélesednek /v. keskenyednek/, éles sarkok eltorzulnak, a megjelenités tengelyeivel párhuzamosnak szánt, de attól minimálisan elhajló élek lépcsőzötté válnak. Mivel a torzitás mérete a digitalizáló "háló"-nak a képre helyezésétől is függ, célszerüen a digitalizálandó képet legfeljebb raszternyi mértékben elmozgatva a digitalizálást érdemes többször is elvégezni és a szemre legjobb ábrát választani a feldolgozás alapjául.

Másik ok, amiért egy fényszedő rendszer nem választhatja a digitalizált képet a betük alapvető ábrázolási módjának, hogy a raszterkép közvetlenül nem alkalmas az itt lévő igényeket kielégitő, méretváltoztatásra. Kicsinyités esetén, maga a kicsinyités elvégzése sem triviális információvesztés nélkül, mig nagyitás esetén ezen kivül a határvonalak is

durvulnak. Másrészt, ha a betűfontokat a szükséges összes méretben külön-külön digitalizáljuk és tároljuk, meredeken növekszik a feldolgozási idő, és a szükséges tárkapacitás.

3.2. MÓDSZEREK FONTOK KÉPEINEK TÁROLÁSÁRA

Az alábbiakban két módszert fogunk összehasonlitani a következő szempontok szerint:

- a várható tömöritési hatásfok
- alkalmasság alak /tipus/ és méretváltozatok képzésére
- érzékenység különböző fontok alaki eltéréseire

Nem vizsgáljuk azonban a tömörités előállitására, illetve a teljes kép visszaállitásához szükséges időigényt.

a/ A "quadtree"-módszer

b/ Görbék illesztése.

/Magáról a görbeillesztésről - technikájáról és kritériumairól - a következő fejezetben részletesen lesz szó./

A "quadtree"-módszer az ábrát négyzetekkel fedi le. Eltárolásra a négyzet oldalhossza, kijelölt csucsának helye kerül. [6], [7], [8], [9]. Nyilvánvaló, hogy tagolatlan határvonalu alakzatok esetén hatékony a módszer. Tekintettel arra, hogy a karakterek zömmel görbevonalakkal határoltak, betük kódolásánál nem várható a módszertől jó hatásfok, becsléseink szerint a lineáris tömöritéseknél nem müködik jobban. Előnytelen tulajdonsága még, hogy hasonlóan a bittérképes ábrázoláshoz rosszul viseli a transzformációt. Megjegyzendő azonban, hogy a képfeldolgozás irodalma sokat foglalkozik a módszerrel, a vele kapcsolatos konverziókra számos algoritmust közölnek. [9], [1o], [11]. Elképzelhető a módszernek egy olyan módositása, amikor nem csak négyzettel, hanem néhány különböző alakzattal fedjük le az ábrát.

A görbéket kétféleképpen illeszthetjük a betükhöz:

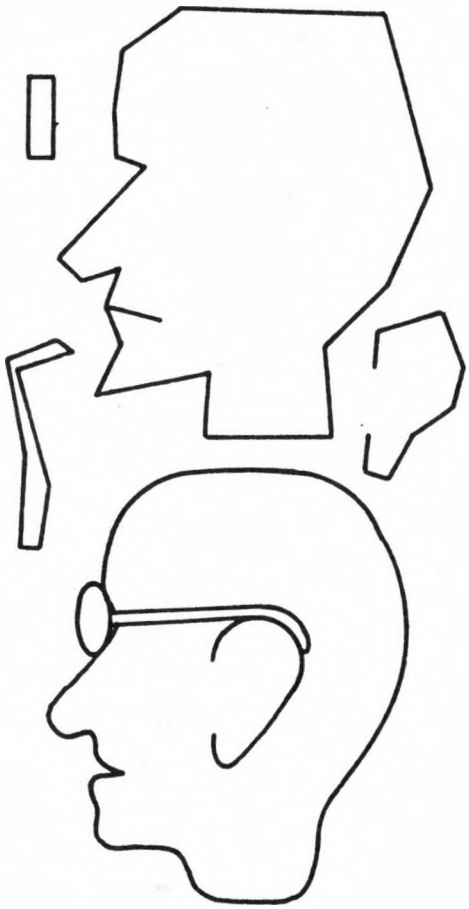
e1/ Gerincvonal módszer: a betükép középvonalához illesztjük a görbét, ugyanakkor minden fonthoz meghatározunk egy tollhegyet is. A gerincvonalon végighuzva a tollhegyet rekonstruálható a teljes betü. Ezt használja D.E. Knuth a METAFONT betütervező rendszerében, lásd [27].

e2/ Határgörbe módszer: a görbéket a betükép határvonalához illesztjük, majd a két kontur közti területet valamilyen kitöltő eljárással kitöltjük.

A fentiek alapján ugy becsüljük, hogy a gerincvonalas leíráshoz mintegy feleannyi pont meghatározására van szükség, mint a határvonalas módszerhez. A határgörbés leirás, mivel több a szabadságfoka, változatosabb alakok leirását teszi lehetővé. A gerincvonalas leirás várhatólag javitható azzal, hogy a tollhegyet változó vastagsággal huzzuk végig a gerincvonalon. /Ez persze a leiráshoz szükséges adatmennyiséget növeli./

Tömöritési hatásfok:

Az idevonatkozó becsléseinket két példára alapozzuk. A 2. ábrán látható fejet [4] 42 pontjával határozták meg. Ebből a fejvonal meghatározásához 21, a szemüveg és fül meghatározásához további 21 pont volt szükséges. Pavlidis módszerével egy általánosnak mondható "g" betüt 32 pontjának megadásával tudunk rekonstruálni. Mivel ez az egyik legbonyolultabb rajzolatu betü, feltétlen felső becslést kapunk egy font kódolására, ha minden karakterhez ugyancsak 32 pontot feltételezünk. 120 különböző karaktert és 2 byte-os számábrázolást feltételezve ez körülbelül 16 kbyte. Ez az érték kb. 16%-a a 3mm x 4mm-es betübélyeg bittérképes ábrázolásának.



2. ábra

Fej közelitése törött vonallal

Érzékenység tipus és méretváltozatokra:

Mint tudjuk, az egyazon fonthoz tartozó különböző méretü karakterek két-három alaprajzból nagyitással illetve kicsinyitéssel nyerhetők. A szokások 1:2 arányu nagyitást engednek meg lineáris növeléssel. A betük döntése szintén lineáris transzformációval történik. Tekintettel a transzformációk linearitására kódoláskor elegendő a transzformációra utaló néhány infor-

mációt kódolni.

Érzékenység alakváltozatokra:

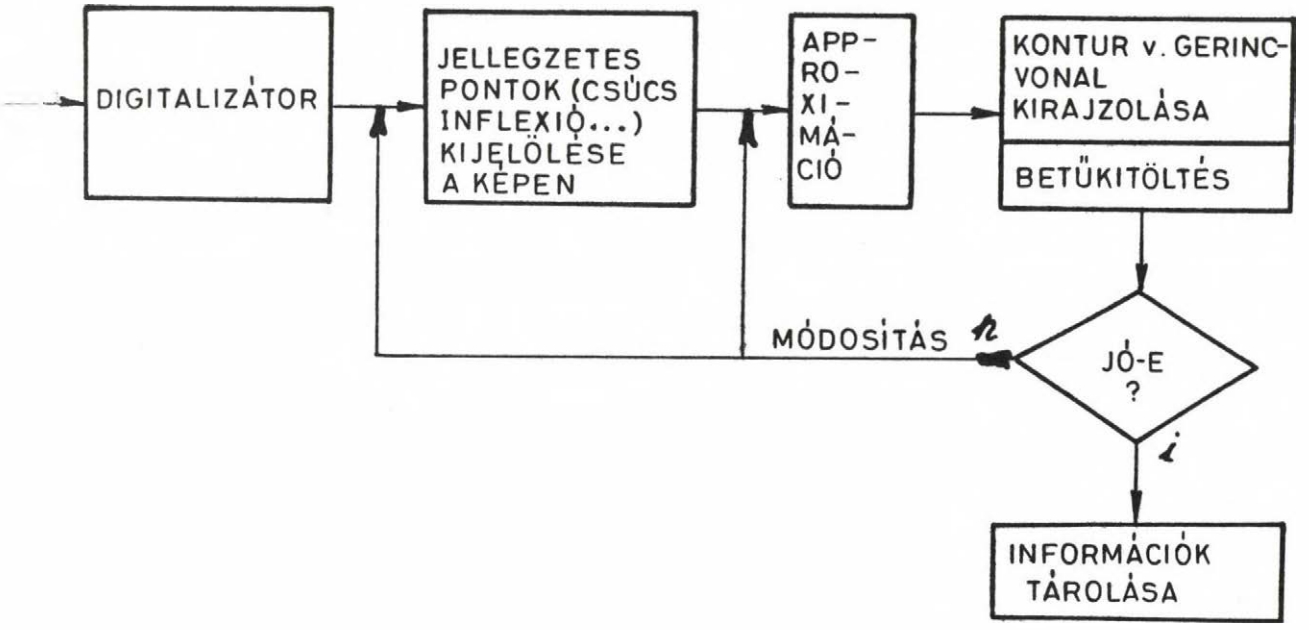
Itt azt vizsgáljuk, hogy különböző fontok kódolásához mekkora helyigénykülönbség szükséges. A fontok különbségét a következők adják:

- a karakterek egy-egy fontban kerekdedebbek, zömökebbek, hossz- és keresztirányu sávjaik aránya változó. A görbeillesztéses kódolásnál ez lényegében nem jár adatnövekedéssel.
- a fontok egymáshoz képest alaki torzulást szenvednek /körökből ellipszisek függőleges egyenesekből ferde egyenesek lesznek stb./. Ennek adatigénynövekedése kevesebb, mint az eredeti 30%.
- végül, ha egy fontban plusz diszitő elemeket - cirádákat, ékezeteket - alkalmaznak, ennek adatigénynövekedése ugy becsülhető, mint a fonté.

Összefoglalva: Mindezen kedvező tulajdonságait figyelembevéve a görbékkel való közelitést tartjuk a legkedvezőbbnek. Arra is jónak látszik a görbékkel való közelités, hogy betükarakteren tul nyomtatványokban használt egyéb jeleket /menetrend, diszités, ... stb./ illetve kottákat irjunk le vele.

4. A BETÜK KÖZELÍTÉSÉRE SZOLGÁLÓ MÓDSZEREK ÁTTEKINTÉSE

4.1. A BETÜRAJZOLÓ VÁZLATOS MÜKÖDÉSE



3. ábra

Betürajzoló folyamatábrája

Ennek az ábrának megfelelően a betürajzoló outputja: "néhány jellegzetes görbepont + további speciális információk" /pl. a közelités milyenségére /B-spline, Bernstein-Beziér, Coons stb./, valamint az esetleges kicsinyitésre, nagyitásra, döntésre és zömitésre vonatkozóak/. Ha tehát a levilágitó-berendezés ugyanazzal a "görberajzoló" és "tartománykitöltő" algoritmussal dolgozik mint a betürajzoló, akkor a levilágitóban semmiféle további ellenőrzésre már nincs szükség.

Az eddig elmondottaknak megfelelően a betürajzolóval kapcsolatos problémákat az alábbi felépitésben tekintjük át:

- Approximációs és interpolációs módszerek általában.
- Néhány speciális görbeillesztő módszer.
- Görberajzoló algoritmusok.
- Tartománykitöltő algoritmusok.
- Összefoglaló megjegyzések.

4.2. APPROXIMÁCIÓS ÉS INTERPOLÁCIÓS MÓDSZEREK ÁLTALÁBAN

A számitógépes grafikában természetesen vetődik fel az u.n. "adatillesztési" probléma, t.i. néhány adott pontra görbét /felületet/ kell illesztenünk. Ez nem más, mint a klasszikus interpolációs probléma. Ha az eredményül kapott görbétől /felülettől/ nem kivánjuk meg, hogy illeszkedjék az adott pontokra, a némileg általánosabb approximációs problémához jutunk. Matematikailag ez a következőt jelenti:

Az adott $f(t)$ /$t \in I$, I valamilyen intervallum/ függvényt egy

$$g(t) = \sum_{i=1}^{n} c_i \psi_i(t)$$

alaku véges lineáris kombinációval kell megközelitenünk, ahol a $\psi_i(t)$-k előre adott tipusu "egyszerübb" függvények /pl. polinomok, trigonometrikus függvények stb./, és mi a c együtthatókat kivánjuk ugy megválasztani, hogy a $g(t)$ valamilyen /később pontositandó/ értelemben "jól" közelitse $f(t)$-t. Általában olyan lesz ez a feltétel, hogy a "legjobban közelitő" $g(t)$ függvény c együtthatói egyértelmüen /pl. lineáris egyenletrendszerrel/ lesznek számolhatóak. A szokásos jelölésekkel legyen $C[a,b]$ az $[a,b]$ zárt intervallumon folytonos, $C^i[a,b]$ pedig az ugyanitt i-szer folytonosan differenciálható

függvények halmaza. Legyen az $f(t) \in C[a,b]$ függvény normája /Csebisev-norma/

$$\|f\| == \max_{a \leq t \leq b} |f(t)| .$$

A "jó" közelitéshez mindenesetre szükséges, hogy $\|f-g\|$ kicsiny legyen. Az alábbi plusz megkötések a jó közelités fogalmának finomitásán tul általában a c_i együtthatók egyértelmü meghatározását is lehetővé teszik. A leggyakrabban használt ilyen további megkötések a következők:

A/ Interpolációs feltevések:

$$g(t_i) = f(t_i) \quad i = 1,\ldots n$$

azaz az approxmáló függvény néhány adott pontban ugyanazt az értéket kell fölvegye, mint a közelitendő függvény.

B/ Vegyes interpolációs és simasági megkötések:

a/ $g(t_i) = f(t_i) \quad i = 1,\ldots k(<n)$

b/ $g'(t_1) = f'(t_1)$ & $g'(t_k) = f'(t_k)$

c/ $g(t) \in C^2[a,b]$.

C/ Ortogonalitási megszoritások:

$$(f-g,\psi_i) = \int_a^b [f(h)-g(t)]\cdot\psi_i(t)dt = 0$$

azaz a $C[a,b]$-n szokásos skaláris szorzat szerint f-g merőleges kell legyen a ψ_i alapfüggvények mindegyikére.

D/ Variációs feltétel:

$$\|f-g\| = \min\{\|f-h\| : h = \sum_{i=1}^{\ell} d_i\psi_i\},$$

azaz a $\psi_1 \ldots \psi_n$ függvények lineáris kombinációi közül, a Csebisev-norma szerint g közeliti meg legjobban f-et.
Már most érdemes megjegyezni azt, hogy a betütervezés és betürajzolás során az "alakhüség" a lényeges szempont, tehát azt, hogy ezek közül a feltételek közül a mi esetünkben melyik hasznos, melyik nem, csak konkrét kisérletezéssel dönthetjük el.
Az esetek legnagyobb részében, még a függvénygörbéket sem az $y = y(x)$ explicit alakban, hanem az

$$x = x(t)$$

$$y = y(t)$$

paraméteres alakban használjuk. Az "alak megtartásán" kivül ez a másik ok, amiért a "görberajzoló" algoritmusok igen fontosak számunkra.

A továbbiakban az általánosan ismert és használt approximációs módszereket tekintjük át, igy a Lagrange- és Hermite-interpolációt, a B-spline-okkal való közelitést, valamint a Bernstein-polinomokkal történő Beziér-féle approximációt.
A majdani felhasználásra gondolva különös figyelmet forditunk a kevés $\psi_i(t)$ alapfüggvénnyel /pl. alacsony, legfeljebb harmadfoku polinomokkal/ való közelitésekre. A tapasztalat ugyanis azt mutatja, hogy az ezekhez szükséges adatok tárolása, az approximációs program futási ideje és a kapott közelités finomsága a legtöbb gyakorlati esetben megfelelő. A gyakorlat azonban ezen módszerek esetleg előforduló "alaktorzitó" hatását is kimutatta /pl. extra inflexió létrehozása, a görbe tulzott kisimitása, indokolatlan oszcilláció produkálása,.../, igy a megfelelő helyeken ezeket a negativumokat is megemlitjük, továbbá néhány olyan módszert is ismertetünk /Akima lokális módszere

[14], spline-ok "feszültség alatt" [18], stb./, amelyek éppen ezeknek a negativumoknak a kiküszöbölésére jöttek létre. Az approximációs módszerekkel foglalkozó részt egy összehasonlitó értékelés zárja, az alábbi munkák alapján [4], [14], [15], [18] .... .

4.3. LAGRANGE-INTERPOLÁCIÓ

Legyen adott n+1 különböző valós szám, $x_o<x_1<...<x_n$ és n+1 darab függvényérték: $y_o,...,y_n$. Ekkor a megfelelő Lagrange-polinomra, a $p_n(x)$ polinomra:

$$\deg(p_n) = n, \quad p_n(x_i) = y_i, \quad i = 0,1,...,n.$$

Jólismert tény, hogy $p_n(x)$ előállitható,

$$p_n(x) = \sum_{i=o}^{n} y_i \cdot L_{i,n}(x)$$

alakban, ahol az $L_{i,n}(x)$ függvények az u.n. Lagrange-féle alapfüggvények, ld. pl. [4,32,33].

A módszer hátránya, hogy egyenletesen választott osztópontok esetén /ami a konkrét esetben a tárolási igény minimalizálása szempontjából ésszerünek látszik/ magasabbfoku görbék /legalább ötödfoku/ az interpolációs görbe hullámzását fokozzák, sőt még az sem biztos, hogy igy a felosztás minden határon tul való finomitásával pontonként tudunk konvergálni a megközeliteni kivánt f(x) függvényhez. Ez az u.n. Runge-Méray jelenség. Szemléletesen a következőre kell gondolnunk: Tegyük föl, hogy f(x) tartalmaz egy vizszintes szakaszt, ekkor a $p_n(x)$ Lagrange-féle interpolációs függvényekre az egyenes szakasz helyett "hullámzást" tapasztalhatunk. Erre a természetes megoldás az lenne, hogy az [a,b] intervallumot kisebb részintervallumokra bontjuk és szakaszonként alacsonyabb foku Lagrange-

-féle interpolációs polinomot veszünk. Az összeállitott interpolációs függvény azonban nem biztos, hogy differenciálható lesz.
Ezt a problémát simasági kikötésekkel tudjuk megszüntetni, de akkor már a Hermite-féle interpolációhoz jutunk.

4.4. AZ HERMITE-FÉLE INTERPOLÁCIÓ

Legyen adott egy f(t) függvény és n+1 csomópont: $a = t_o < t_1 < \ldots < t_n = b$. Az ezen beosztáshoz tartozó harmadfoku Hermite-féle interpolációs $p_{3,i}(t)$ polinomokra az alábbiak teljesülnek:

$$p_{3,i}(t_{i-1}) = f(t_{i-1}) \qquad p'_{3,i}(t_{i-1}) = f'(t_{i-1})$$

$$p_{3,i}(t_i) = f(t_i) \qquad p'_{3,i}(t_i) = f'(t_i)$$

és a $p_{3,i}(t)$ függvényekből egy /szakaszonként harmadfoku/ $s(t) \in C^1[a,b]$ függvény rakható össze. Az Hermite-interpolációról lsd. pl. [4,32,33]. Persze az első deriváltak helyett magasabbrendü deriváltakra is tehetünk hasonló megszoritásokat, s igy ugyan elvileg akármilyen simaságot elérhetünk, de ehhez szükségünk van /a harmadfoku esetben is/ a belső pontokban számitható deriváltak ismeretére. A mi esetünkben ez elég nehézkesen tehető meg /pláne kellemetlen lenne a magasabbrendü deriváltakat megadni/, igy olyan módszer lenne kellemes, amelynél az approximációhoz csak a szélső pontokban kell ismernünk a deriváltakat.
/Ez a kikötés elég természetes a betük esetében, t.i. az egyes szakaszok végpontjait ugy fogjuk választani, hogy ott az érintő vizszintes vagy függőleges legyen, illetve az illető pont csucs, aholis az érintőt tényleg kell ismernünk./
Ezt a Hermite-interpoláció egy Coons-tól származó speciális esete /lsd. [4],[12] megteszi, és valóban a Coons-interpoláció

a számitógépes grafika egyik gyakran használt eszköze. A módszert később, a harmadfoku görbékkel való közelitések részletes elemzésénél tárgyaljuk.

4.5. INTERPOLÁCIÓ B-SPLINE-OKKAL

A spline függvények igen kedvezőek interpolációs feladatok megoldására, mert egyesitik a polinomok egyszerü kezelhetőségét egy általunk felépitett simasággal. A hatvanas évektől kezdve intenziven tanulmányozták a spline függvények tulajdonságait, több monográfia és kézikönyv is született ebben a témakörben [3],[3o],[31]... .
Mi itt csak a polinomiális spline függvények /a B-spline-ok/ legérdekesebb és leghasznosabb tulajdonságait gyüjtjük össze.

Legyenek adottak az $x_o < x_1 < \ldots < x_n$ valós számok. Egy S(x) m-ed foku, $x_o \ldots x_n$ csomópontu spline függvény olyan egyváltozós függvény, amelyre teljesül az alábbi két tulajdonság:

1/ Minden (x_i, x_{i+1}) intervallumban S(x) legfeljebb m-ed foku polinommal adható meg. /Itt $x_{-1} = -\infty$ és $x_{n+1} = +\infty$ /.

2/ S(x) és 1,2,...m-1 rendü deriváltjai az egész számegyenes folytonosak.

Vegyük észre, hogy $m = 0$ esetén egy spline függvény nem más mint egy lépcsősfüggvény és, hogy $m > 0$-ra a legfeljebb m-ed foku spline függvények éppen a lépcsősfüggvények m-szeres határozatlan integráljai.

Mivel ez a definició megengedi, hogy akár az egész számegyenesen ugyanaz a polinom definiálja S(x)-et, igy a spline-okkal való approximáció magában foglalja a polinomokkal történőt.

harmadfoku spline-ok /és itt jegyezzük meg, hogy az előző szakaszban már emlitett Coons-féle approximációval kapható függvények is spline-ok/ a gyakorlati görbeillesztési problémák megoldásának egyik leghasznossabb eszközei. A gyakorlat céljaira elegendő egy speciális spline tipust definiálni. Nevezetesen legyen

$$x_+^r = \begin{cases} x^r & , \text{ ha } x > 0 \\ 0 & , \text{ ha } x <= 0 \end{cases}$$

Egy (2k-1)-ed /páratlan!/ foku S(x) spline az $x_0,...x_n$ csomópontokkal *természetes* spline, ha a $/-\infty, x_0/$ és az $/x_n,+\infty/$ intervallumokon kisebb mint k-ad foku polinommal adható meg. Megmutatható, hogy a természetes spline-ok a görbeillesztés problémájának egyértelmü megoldását szolgáltatják, méghozzá a lehető legsimábbat abban az értelemben, hogy minimalizálják az

$$\int_a^b [s^{(k)}(x)]^2 dx, \quad (0 < k < n)$$

integrált!

Visszatérve az

$$s'(t_0) = f'(t_0),$$

$$s(t_i) = f(t_i), \qquad t = ih \quad i=0,...n$$

$$s'(t_n) = f'(t_n)$$

egyenletes felosztásu interpolációs problémára, a természetes spline-okkal való közelités a következőképpen mehet /lsd. [29]/.

Legyen

$$\beta_r(t) = \frac{1}{r!} \sum_{j=-k}^{+k} (-1)^{j+k} \binom{2k}{j+k} (j-t)_+^r .$$

Könnyen ellenőrizhetőek β_r alábbi tulajdonságai:

$$\beta_r(t) > 0, \quad \text{ha} \quad |t| < k$$

$$\beta_r(t) = 0, \quad \text{ha} \quad |t| = k$$

$$\beta_r(t) \equiv 0, \quad \text{ha} \quad |t| > k$$

$\beta_r(-t) = \beta_r(t)$ és $\beta_r(0) > \beta_r(t)$, továbbá

$$\sum_{i=-\infty}^{+\infty} |\beta_r(i)| = \sum_{i=1-k}^{k-1} \beta_r(i) = 1.$$

Mivel a $\beta_r(t)$ függvények deriváltjai a

$$\beta_r^{(p)}(t) = \frac{1}{(r-p)!} \sum_{j=-k}^{+k} (-1)^{j+k+p} \binom{2k}{j+k} (j-t)_+^{r-p}$$

képlettel számolhatóak, kapjuk, hogy $\beta_r(t)$ (r-1)-szer folytonosan differenciálható az egész számegyenesen, valamint

$\beta_r^{(p)}(+k) = 0.$

Tekintsük most a [o,1] zárt intervallumon a $h = 1/q$ sürüségü egyenletes csomópontrendszert, azaz legyen $t_j = jh$, ahol $j = 0,1,\ldots,q$.

Megmutatható, hogy a $\beta_r(qt-i)$ $i = 1-k, 2-k,\ldots,q+k-1$ függvényrendszer lineárisan független a C[0,1] valós vektortérben, azaz egy r-edfoku spline /a t_j csomópontokkal/ felirható, mint

egy

$$s_r(t) = \sum_{i=1-k}^{q+k-1} a_i \beta_r(qt-i)$$

lineáris kombináció. Mivel a $\beta_r(qt-i)$ függvények kompakt tartójuak, minden fix $t\in[0,1]$-re ez tulajdonképpen csak $(r+1)$-tagu összeg. Továbbá

$$\sum_{i=1-k}^{q+k-1} \beta_r(qt-i) = 1$$

és igy

$$\|s_r(t)\| < \sup_i |a_i| \text{ minden } t\in[0,1]\text{-re.}$$

Könnyen számolható, hogy

$$\frac{d^p}{dt^p} s_r(t) = q^p \sum_{i=i-k}^{q+k-1} a_i \beta_r^{(p)}(qt-i).$$

Ennek alapján egy tetszőleges B/ tipusu approximációs probléma megoldása olyan lineáris egyenletrendszerek megoldásával tárgyalható, amelyek mátrixa a $\beta_r^{(i)}(j)$ értékekből épül fel, kedvező esetben nem-szinguláris, és igy a vegyes interpolációs és simasági kikötések egyértelmü kielégitését teszi lehetővé. Ilyen például a $k = 2$ eset, t.i. a harmadfoku B-spline-ok esete, amelyet a későbbiekben részletesen megvizsgálunk.

A B-spline-ok alkalmazásának egyik fő előnye "lokalitásuk"-ban rejlik. Ha a $\underline{c} = (f'(t_o), f(t_o), f(t_1), \ldots, f(t_i), \ldots, f(t_n), f'(t_n)$ sorvektor valamely c_i komponensét megváltoztatjuk, akkor legfeljebb 4 darab a együttható értéke változik meg. Schönberg - aki először tárgyalta a B-spline-okat - megmutatta, hogy a B-spline-ok egyértelmüen meghatározottak, mint a legkisebb tartóju spline-függvények.

4.6. GÖRBÉK BÉZIER-FÉLE APPROXIMÁCIÓJA

Az eddigi eljárások mind interpolációs jellegüek voltak, azaz néhány pontban megegyeztek a közelitendő függvénnyel és azt /az A),B),C) ill. D) feltételek értelmében/ jól közelitették. Most egy más jellegü módszert ismertetünk, ti.i az adott pontokra nem illesztjük rá a görbét, csupán megközelitjük. Az ilyen tipusu eljárások akkor érdekesek számunkra, ha az illesztés pontosságánál fontosabb, hogy "alakjában" hasonlóan közelitsük jól a görbét. Ugyancsak előtérbe kerül ez a módszer, ha a közelitésre tett kirovásokat nem tudjuk az A/, B/, C/ és D/ feltételekhez hasonló módon matematikai alakban is megfogalmazni.
A módszer különösen autó, repülő és hajóépitési tervezéseknél bizonyult hasznosnak, ahol nincsen jól definiálhatóan legjobb illesztés, hanem az illesztés jóságát a tervező birálja felül. A Bézier-módszer talán a leghatékonyabb olyan interaktiv módszer, amely "alakhü" approximációt tud megvalósitani.

Összefoglalva tehát a Bézier módszer két logikailag jól elkülönithető részből áll:

egyik a most ismertetendő approximációs eljárás a Bernstein-polinomokkal, másik egy interaktiv grafikai ötlet, az u.n. kontrollpoligon használata, amely a rajzolandó görbe vizuális elképzelését könnyiti meg. Ez a grafikai ötlet az approximációs módszer megválasztásától tulajdonképpen független, Bernstein-polinomok helyett B-spline-okra is müködik, sőt a következő fejezetben Pavlidis módszerében is előjön.

Bézier módszere a Bernstein-polinomokkal való approximációra épül és a következőképp irható le:

Legyen $f(t) \in C[0,1]$ tetszőleges. Az f-hez tartozó n-edfoku Bernstein-polinomot a

$$B_n(f(t) = \sum_{k=o}^{n} f(\frac{k}{n}) \phi_{k,n}(t)$$

képlet adja meg, /lsd [3]/ ahol

$$\phi_{k,n}(t) = \binom{n}{k} t^k (1-t)^{n-k} \quad k = 0,1,\ldots,n$$

a binomiális eloszlásu valószinüségi változó sürüség függvénye. A valószinüségszámitásból jól ismertek $\phi_{k,n}$ alábbi tulajdonságai:

1. $\phi_{k,n} > 0$ és $\sum_{k=0}^{n} \phi_{k,n}(t) = 1$

2. $\phi_{0,n}(0) = 1$

$$\phi_{k,n}^{(k)}(0) = \frac{n!}{(n-j)!}$$

$$\phi_{k,n}^{(p)}(0) = 0, \quad \text{ha} \quad p = 0,1,\ldots,k-1 \qquad \text{és}$$

$$\phi_{k,n}^{(q)}(1) = 0, \quad \text{ha} \quad q = 0,1,\ldots,n-k-1 \quad .$$

3. $\phi_{n,n}^{(p)} = 0 \quad , \quad \text{ha} \quad p = 0,1,\ldots,n-1$

$$\phi_{n,n}(1) = 1.$$

4. $$\phi_{k,n}^{(n-k)}(1) = \frac{(-1)^{n-k} n!}{(n-k)!}$$

5. $$\phi_{k,n}\left(\frac{k}{n}\right) = \binom{n}{k} k^k (n-k)^{n-k} > \phi_{k,n}(t), \quad \text{ha} \quad t \neq \frac{k}{n}.$$

A feltételek azt mutatják, hogy a B_n Bernstein-polinomok általában csak az f(0) és f(1) görbepontra illeszkednek, a többit csak megközelitik.

A $B_n(t)$ függvények végpontbeli deriváltjaira:

$$\frac{d^p}{dt^p} B_n(f(t))\Big|_{t=o} = \frac{n!}{(n-p)!} \sum_{k=o}^{p} (-1)^{p-k} \binom{p}{k} f\left(\frac{k}{n}\right)$$

$$\frac{d^p}{dt^p} B_n(f(t))\Big|_{t=1} = \frac{n!}{(n-p)!} \sum_{k=o}^{p} (-1)^{k} \binom{p}{k} f\left(\frac{n-k}{n}\right) .$$

Nagyon nevezetes tény, hogy a Bernstein-polinomokra érvényes Weierstrass approximációs tétele, nevezetesen n növelésével $B_n(f(t))$ egyenletesen konvergál f(t)-hez. Továbbá, ha az oszcillációt egy adott szakasz átmetszéseinek számával mérjük, akkor f-nél magánál is simábban közeliti őt a Bernstein polinomja. A Bernstein-polinomok a Csebisev-norma szerint meglehetősen lassan közelitenek f(t)-hez, de mint Bézier megmutatta, sima szabadkézi rajzok közelitésére kiváló eszközök. Bézier módszeréhez meg kell adnunk egy poligon csucsait, az összekötés sorrendjét, és minden csucshoz egy-egy valós számot, azaz egy poligont, melynek csucsaira az $(\frac{i}{n}, v_i)$ szimbólumokat irjuk, ahol az illető csucs a poligon i-edik csucsa, a v_i pedig az emlitett valós szám. Egy ilyen π cimkés poligonhoz tekintsük a

$$B_n(\pi,t) = \sum_{k=o}^{n} v_k \, \phi_{k,n}$$

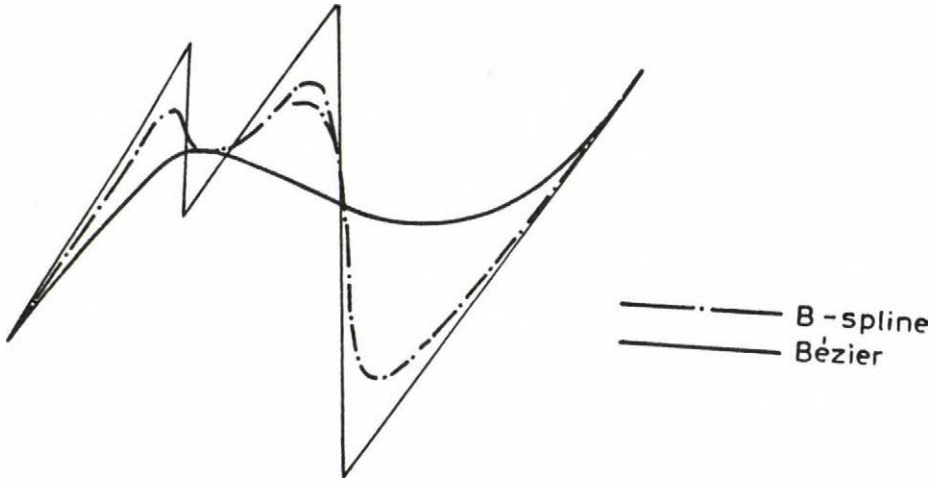
Bernstein-polinomot.

A Bézier-módszerben a $B_n(\pi,t)$-vel jelölt Bézier-görbe átmegy a poligon végpontjain és érintője a végpontokban a poligon első illetve utolsó oldalegyenese. A Bézier-görbe egészen a Bézier-poligon konvex burkában fekszik, és "meglehetősen" jól tükrözi ezen poligon tulajdonságait. Tulajdonképpen

ez a módszer egy interaktiv görbeközelitési segédeszköz, amely nemcsak eredeti alakjában, t.i. a Bernstein-polinomokkal, hanem tetszőleges más alapfüggvényekkal is alkalmazható.

Például a B-spline függvények, mint alapfüggvények a következő előnyökkel járnak:

1. A közelitő görbe jobban simul az alappoligonhoz. /Ez megkönnyiti a Bézier-poligon kijelölését./
2. A B-spline-ok "legkisebb tartóju"-sága miatt a B-spline-okkal való approximáció lokális, azaz egy csucs változtatásával csak kevéssé változik az approximáló görbe.
3. A Bernstein-polinomokkal való approximációnál, ha a nagyobb pontosság érdekében növeljük a Bézier-poligon oldalainak számát, a polinom foga egyre nő, mig a B-spline-ok használata esetén ez tetszőlegesen megválasztható.



4. ábra

B-spline és Bézier poligon-os közelités

4.7. A FOURIER-APPROXIMÁCIÓ

A periódikus, folytonos függvények Fourier-sorba fejtéséből is lehet approximációs módszert nyerni. Ha az f függvény csak az $0 = x_o$ <...> $x_n = 2\pi$ pontokban adott a hozzátartozó y_i függvényértékekkel, és $y_n = y_o$, akkor

$$a_o + \sum_{i=1}^{n} (a_i \cos ix + b_i \sin ix) = g(x)$$

alakban lehet megközeliteni az f függvényt. Ez is lineáris egyenletrendszer megoldására vezet, ezt azonban technikailag viszonylag nehéz kiszámitani. Csak arra az esetre jó, ha a megközelitendő függvényről előre tudjuk, hogy igen hullámos /pl. hullámjelenségeknél stb./. Esetünkben, betüközelitésnél valószinütlen, hogy ez a módszer használható.

5. SPECIÁLIS KÉRDÉSEK

5.1. TÖRÖTTVONALLAL VALÓ KÖZELITÉS

A raszterfelbontás szükségszerü durvasága miatt minden görbe egyenes szakaszokból áll. Természetesen megtehetjük, hogy csupán ezen szakaszok végpontjait kódoljuk. Azokon a részeken, ahol a betüalak erősen görbül, szinte minden rasztersávba kerül egy ilyen poligonnak csucspontja, igy tárkapacitásigénye meglehetősen nagy. Tulajdonképpen ez a következő elemi kódolási módszernek, az u.n. lánckódos módszernek felel meg:
Adjunk meg egy O kezdőpontot és "vezessük a ceruzát" ettől kezdve az alábbi szabályok szerint:

le: OO /Dél/

fel: 11 /Észak/

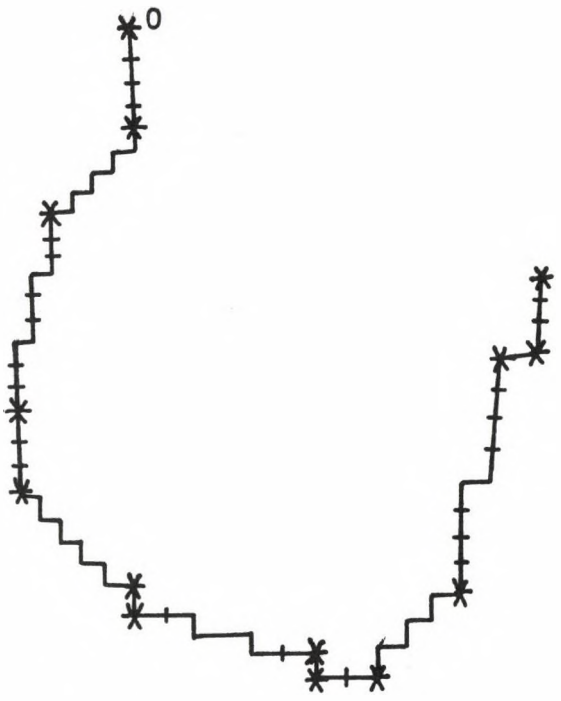
balra: 1O /Nyugat/

jobbra: O1 /Kelet/

/Persze ehelyett 3 bitben ábrázolható E,D,K,NY,EK,ENY,DK,DNY 8 irány segitségével is vezethetjük a ceruzát./

Egy ilyen elemi tömöritès helyigénye kb. a zárt görbét befoglaló téglalap kerülete /raszterpontszámban mérve/ szorozva 2 bittel. A konkrét esetben pl. a töröttvonallal való megadáshoz a *-gal jelölt pontokat kell megadni, ami sok pont megadását jelenti ugyan, de a koordináta-megváltozások kevés bitben ábrázolhatók.

Ennek a módszernek az az előnye, hogy az egyenes szakaszok egyszerüen rajzolhatóak /lsd pl. később a Bresenham féle algoritmust stb./, hátránya viszont az alak és méretváltozásra való érzékenység.



5. ábra

Töröttvonalas közelités

5.2. MÁSODFOKU GÖRBÉKKEL VALÓ KÖZELITÉSEK

Ebben a szakaszban két módszert tárgyalunk. Az elsőnek az az érdekessége, hogy egy fényszedő levilágitó-berendezés karaktergenerátora számára fejlesztette ki M. Tervonen, H. Hakalahti és P. Lappalainen /lsd. [17]/. A karakter kontur-vonalát közelitik egyenes szakaszok és ellipszis-ivek alkalmazásával.

Az egyenes szakaszokat az

$$y = y_1+k(x-x_1), \quad k = (y_n-y_1)/(x_n-x_1)$$

képlettel irják le és az egymást követő pontokat az

$$y_{n+1} = y_n + k \qquad (n = 1,2,\ldots,(x_n - x_1)$$

rekurzió segitségével generáljuk. Igy az egyenes szakaszokat /a nagyobb sebesség eléréséért kissé redundánsan/ az x_1, x_n, y_1, y_n és k értékek tárolásával kezelik.

Az (x_o, y_o) középpontu a ill. b kis és nagytengelyü ellipszis egyenlete:

$$\frac{(x-x_o)^2}{a^2} + \frac{(y-y_o)^2}{b^2} = 1$$

Ennek az ellipszisnek az (x_1, y_1) és (x_2, y_2) pontja közti ivét a következő rekurziv módon irhatjuk le:

$$\frac{((x_{1+n})-x_o)^2}{a^2} + \frac{(y_{n+1}-y_o)^2}{b^2} = 1$$

$$\frac{(x_{1+n-1})-x_o)^2}{a^2} + \frac{(y_n-y_o)^2}{b^2} = 1$$

amiből kivonással kapjuk, hogy

$$(y_{n+1}-y_o)^2 - (y_n-y_o)^2 = -2\left(\frac{b}{a}\right)^2 (x_1-x_o) + \left(\frac{b}{a^2}\right) - 2n\left(\frac{b}{a}\right)^2.$$

Legyen $z_n = (y_n-y_o)^2$. Erre az előző szerint a

$$z_{n+1} = z_n - 2\left(\frac{b}{a}\right)^2 (x_1-x_o) + \left(\frac{b}{a}\right)^2 - 2n\left(\frac{b}{a}\right)^2$$

rekurzió teljesül. Igy /a $\Delta Z_{n+1} = Z_{n+1} - Z_n$ jelöléssel/

$$\Delta Z_{n+1} = \Delta Z_n - 2\left(\frac{b}{a}\right)^2$$

rekurziót kapjuk, ahol

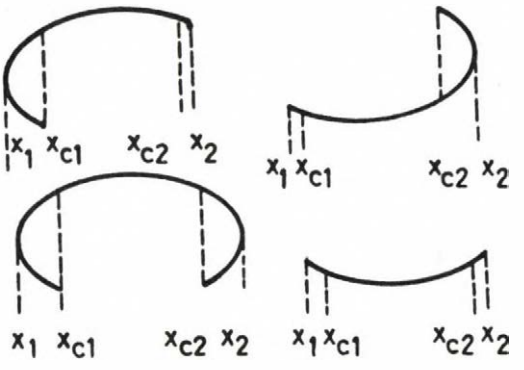
$$\Delta Z_1 = -2\left(\frac{b}{a}\right)^2 (x_1 - x_o) + \left(\frac{b}{a}\right)^2$$

és

$$\Delta Z_2 = -2\left(\frac{b}{a}\right)^2 (x_1 - x_o) - \left(\frac{b}{a}\right)^2$$

és $Z_1 = (y_1 - y_o)^2$.

Az elliptikus ivet az alábbi paraméterekkel adják meg: $x_1, x_2, Z_1, \Delta Z_2, y_o$, $2(b/a)^2$ valamint az alábbi tipusu két iv megkülönböztetésére és a gyorsitásra a további x_{c_1} és x_{c_2}



6. ábra

Elliptikus iv megadása

pontokat. Természetesen olyan karaktergenerátort készitettek, ahol a rekurzióból adódó számolási hiba még megengedhető ha-

tárok között mozog.

A másik módszer kupszeletdarabokkal közelit és T. Pavlidistől származik. Tulajdonképpen a B-spline approximáció és a Bézier approximáció módján közelit szakaszonként kupszeletdarabokkal /igy tehát általánositása a másodfoku spline-okkal történő közelitésnek/ és a közelités interaktivitását a Bézier-approximációnál megismert kontrollpoligon segitségével biztositja. További előnye algoritmusának, hogy olyan eljárást is kidolgozott, amelyek adott görbe kupszeletdarabokkal történő közelitését automatikussá teszik, valamint, hogy kupszeleteket rajzoló algoritmusokat is megad. /ld. majd később/. A kupszeleteket az

$$x(t) = \frac{x_o t^2+x_1 t+x_2}{w_o t^2+w_1 t+w_2}$$

$$y(t) = \frac{y_o t^2+y_1 t+y_2}{w_o t^2+w_1 t+w_2}$$

racionális paraméterezés segitségével kezeli és egy kupszeletdarabot a következőképpen definiál:

a/ V_i: kontrollpoligon csucsai

b/ $0 \leq p_i \leq 1$: ami megadja, hogy a $V_i\ V_{i+1}$ szakaszon hol helyezkedik el a csomópont

c/ $0 \leq q_i \leq 1$: ami megmondja, hogy hol metszi a kupszeletdarab a V_{i+1}-et az őt megelőző és követő csomópont felezőpontjával összekötő szakaszt.

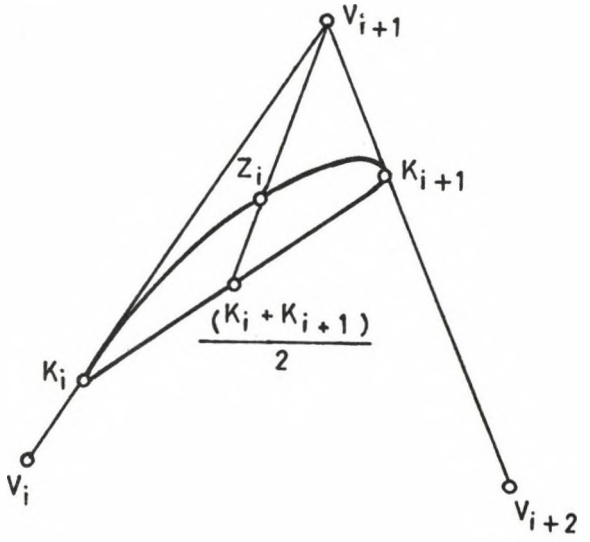
A K_i csomópontok a következőképp vannak definiálva:

$$K_i = (1-p_i)V_i + p_i V_{i+1}$$

és a kontrollpoligon oldalai a csomópontbeli érintőket is meghatározzák. A kupszeletet a

$$Z_i = q_i V_{i+1} + (1-q_i)\,\frac{K_i + K_{i+1}}{2}$$

pont határozza meg.



7. ábra

Kupszeletdarabokkal történő közelítés

Megjegyezzük, hogy a $q \pm 0.5$ választás éppen a parabolákkal történő másodfoku B-spline approximációt adja vissza.

Érdekességként megemlitjük, hogy a görbeközelités automatizálását az teszi lehetővé, hogy egy pont és egy kupszelet távolságára jól kezelhető explicit kifejezések adhatók meg [15].

Pavlidis a következő számitógépes kisérleti eredményeket emeli ki:
Egy-egy karaktert a PDP 11/70 gépen kb. 1 sec, a VAX 11/780 gépen kb. 0.5 sec teljes CPU idő alatt közelitett meg. Egy epicikloidot 15 csomópont felhasználásával tudott jól megközeliteni, mig ugyanezt a problémát harmadfoku spline-okkal 14 és 31 közötti számu csomóponttal tudta megoldani Reeves és Sermer. Ez a feladat /Pavlidis módszerével/ a VAX gépen kb. 1.5 sec-ig tartott. A módszer sarkalatos pontja a Bézier--módszerből ismert kontrollpoligon felvétele, amely ugyanazokkal az előnyökkel és hátrányokkal jár, mint amit a Bézier--módszernél elmondtunk.

5.3. HARMADFOKU GÖRBÉKKEL VALÓ KÖZELITÉSEK

Itt az alkalom a Coons féle approximáció ismertetésére. Ekkor /pl. a térben/ a homogén pont-koordináták segitségével

$$\tilde{P}(t) = (wx(t)\ wy(t)\ wz(t)\ w(t))$$

$$wx(t) = a_3t^3+a_2t^2+a_1t+a_o$$

$$wy(t) = b_3t^3\ b_2t^2+b_1t+b_o$$

$$wz(t) = c_3t^3+c_2t^2+c_1t+c_o$$

$$w(t) = d_3t^3+d_2t^2+d_1t+d_o$$

azaz mátrixalakban

$$\tilde{p}(t) = (t^3t^2t\ 1) \times \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_o & b_o & c_o & d_o \end{pmatrix} \overset{\text{jel}}{=} (t^3t^2t1) \times A$$

és persze

$$\tilde{P}'(t) = (3t^2 \quad 2t \quad 1 \quad 0) \times A$$

$$\tilde{P}''(t) = (6t \quad 2 \quad 0 \quad 0) \times A \ .$$

Ugy kivánjuk az A mátrix elemeit megválasztani, hogy a

$$\tilde{P}(t)\Big|_{t=o} = p_o \qquad \tilde{P}'(t)\Big|_{t=o} = p_o'$$

$$\tilde{P}(t)\Big|_{t=1} = p_1 \qquad \tilde{P}'(t)\Big|_{t=1} = p_1'$$

feltételek teljesüljenek, vagyis

$$\begin{matrix} p_o \\ p_1 \\ p_o' \\ p_1' \end{matrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} \times A.$$

Megoldva ezt az egyenletrendszert

$$A = M \times \begin{pmatrix} p_o \\ p_1 \\ p_o' \\ p_1' \end{pmatrix}, \quad \text{ahol} \quad M = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

azaz explicite

$$P(t) = (2t^3-3t^2+1 \quad -2t^3+3t^2 \quad t^3-2t^2+t \quad t^3-t^2) \times \begin{pmatrix} p_o \\ p_1 \\ p_o' \\ p_1 \end{pmatrix} .$$

Látható, hogy $\tilde{P}(t)$ nem más, mint az $n = 1$, $t_o = 0$, $t_n = 1$ paraméterértékekhez tartozó harmadfoku Hermite-interpolációs polinom. Ha nem 2, hanem (n+1) pont, $(p_o,p_1,\ldots,p_n)$ van megadva, akkor szakaszonként ilyen Coons-féle görbékre bontjuk. Mivel a belső pontokban az érintők nincsenek megadva, az igy nyert szabadságot arra használhatjuk fel, hogy a C^2-simaságot biztositani tudjuk, azaz $j = i+1$-re

$$\tilde{P}_i(1) = \tilde{P}_j(0) = p_j, \quad \tilde{P}_i'(1) = \tilde{P}_j'(0) \text{ és } \tilde{P}_i''(1) = \tilde{P}_j''(0).$$

Mivel ilymódon szakaszonként legfeljebb harmadfoku polinomokkal C-simaságu közelitést tudunk megvalósitani, láthatjuk, hogy a Coons-approximáció a B-spline-okkal való közelitésnek speciális esete.
Iktassuk ide /paraméteres megadásu görbékre/ a Coons féle approximáció egy megoldását:
a kirajzolandó görbe legyen az

$$x = x(t)$$

$$y = y(t)$$

paraméteres alakban megadva. A végpontok legyenek (x_o,y_o) ill. (x_n,y_n), az itteni érintők $(\dot{x}_o,\dot{y}_o)$ ill. $(\dot{x}_n,\dot{y}_n)$, mig a közbülső pontok (x_i,y_i) $(1 < i < n-1)$. Mindkét koordinátafüggvényt, külön-külön, szakaszonként harmadoku függvénnyel közelitjük. Tehát pl. az x koordinátára

$$x = x_i(t) \qquad 0 \leq t \leq 1$$

minden $[x_i, x_{i+1}]$ szakaszon /i = 0,...n-1/ és az $x_i(t)$ harmadfoku függvények csatlakozására az alábbi simasági feltételeket kötjük ki:

$$x_o(0) = x_o$$

$$\dot{x}_o(0) = \dot{x}_o$$

$$x_o(1) = x_1(0) = x_1 \qquad \text{/csatlakozás/}$$

$$\dot{x}_o(1) = \dot{x}_1(0) = \dot{x}_1 \qquad \text{/ezt fogjuk kiszámolni/}$$

$$\ddot{x}_o(1) = x_1(0) \qquad /C^2\text{-csatlakozás/}$$

. . .
. . .
. . .

$$x_i(1) = x_{i+1}(0) = x_{i+1} \qquad \text{/csatlakozás/}$$

$$\dot{x}_i(1) = \dot{x}_{i+1}(0) = \dot{x}_{i+1} \qquad \text{/ezt fogjuk kiszámolni/}$$

$$\ddot{x}_i(1) = x_{i+1}(0) \qquad /C^2\text{-csatlakozás/}$$

. . .
. . .
. . .

$$\dot{x}_{n-1}(1) = \dot{x}_n$$

$$x_{n-1}(1) = x_n \ .$$

Irjuk fel azt a harmadfoku Hermite-polinomot, amelyre

$$\ell(0) = p_o, \ \ell(1) = p_1, \ \ell'(0) = q_o, \ \ell'(1) = q_1.$$

Ezt a

H) $\ell(t) = p_o+(2t^3-3t^2)(p_o-p_1) +$

$$+(t^3-2t^2+t)q_o+(t^3-t^2)q_1$$

képlettel tehetjük meg. Számoljuk ki ℓ''-t

H") $\quad \ell''(t) = (12t-6)(p_o-p_1)+(6t-4)q_o+(6t-2)q_1$

$t = 0$-t és $t = 1$-et helyettesitve

$$\ell''(0) = 6p_1-6p_o-4q_o-2q_1$$

$$\ell''(1) = 6p_o-6p_1+2q_o+4q_1 \ .$$

Ezeknek a képleteknek a beirásával S) a következőképpen alakul:

$$6x_o-6x_1+2\dot{x}_o+4x_1 = -6x_1+6x_2-4\dot{x}_1-2\dot{x}_2$$

$$\cdots$$

$$6x_{i-1} - 6x_i+2\dot{x}_{i-1}+4\dot{x}_i = -6x_i+6x_{i+1}-4\dot{x}_i-2\dot{x}_{i+1}$$

$$\cdots \qquad (i = 1,\ldots,n-1).$$

Itt $\dot{x}_1,\ldots \dot{x}_{n-1}$ $n-1$ darab ismeretlen, a felirt összefüggések pedig $n-1$ darab lineáris egyenletet jelentenek. Átrendezve az egyenleteket

$$4\dot{x}_1+\dot{x}_2 = 3x_2-3x_o-\dot{x}_o$$

$$\dot{x}_1+4\dot{x}_2+\dot{x}_3 = 3x_3-3x_1$$

$$\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$$

$$\dot{x}_{n-2}+4\dot{x}_{n-1} = 3x_n-3x_{n-2}-\dot{x}_n \quad .$$

Erről az egyenletrendszerről jól látszik, hogy egyértelmüen megoldható, mivel mátrixa a /tartópontoktól független!/ tridiagonális mátrix.

$$\begin{pmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & 0 \\ & 1 & 4 & 1 & & \\ & & & & & \\ & 0 & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{pmatrix}$$

5.4. TOVÁBBI SPECIALIS MÓDSZEREK

Ebben a szakaszban két olyan módszert ismertetünk, amelyek bevallottan abból a célból jöttek létre, hogy az előző szakaszokban leirt módszerek hibáit korrigálják. Először lássuk a "feszültség alatti" spline-okat /lásd még CACM 17, 4, Apr 1974 pp. 218-223, ACM alg. No. 476/.
Itt az inflexiós pontok számát fogjuk korlátozni. Ennek megfelelően csak olyan interpoláló függvényeket engedünk meg, amelyekre:

i/ $y \in C^2(a,b)$

ii/ y" legfeljebb egyszer vált előjelet

iii/ az y függvény egyes darabjai csak előre megadott pontok közötti teljes szakaszok lehetnek.

Az egyszerüség kedvéért föltesszük, hogy a felosztás egyenletes, azaz

$$x_i = i \quad \text{és legyen} \quad y_i' = y'(x_i),\ y_i'' = y''(x_i),$$

/ahol megadjuk/. Legyen továbbá

$$d_i = y_{i-1} - 2y_i + y_{i+1} \qquad 1 \leq i \leq n$$

és a végpontokban

$$d_o = \begin{cases} (y_1 - y_o) - y_o' & , \quad \text{ha} \quad y_o' \text{ adott} \\ y_o'' & , \quad \text{ha} \quad y_o'' \text{ adott} \end{cases}$$

$$d_{n+1} = \begin{cases} y_{n+1}' - (y_{n+1} - y_n), & \text{ha} \quad y_{n+1}' \text{ adott} \\ y_{n+1}'' & , \quad \text{ha} \quad y_{n+1}'' \text{ adott} \end{cases}$$

Egy az előző definició szerint megengedett intepoláló függvényre az x^* hely extra inflexiós hely, ha

$$x_i < x^* < x_{i+1} \quad d_i d_{i+1} > 0 \qquad 0 \leq i \leq n$$

vagy

$$x^* = x_i \qquad 1 \leq i \leq n. \quad \text{/tehát } x^* \text{ nem végpont!/}$$

A módszer lényege az extra inflexiós pontok kiküszöbölése. Ehhez az ad segitséget, hogy az extra inflexiós pontok nemlétezésének szükséges és elégséges feltétele, hogy

$$y_i''d_i > 0, \quad \text{ha} \quad y_i'' \neq 0 \qquad i = 0,\ldots,n+1$$

$$y_{i+1}''y_{i-1}'' > 0, \quad \text{ha} \quad y_i'' = 0 \qquad i = 1,\ldots,n.$$

Ezekre a célokra a következőket javasolja Schweikert [18]. Legyen

$$y(x) = f_i(x), \quad \text{ha} \begin{cases} x_i \leq x < x_{i+1}; & i = 0,1,\ldots,n-1 \\ x_n \leq x \leq x_{n+1}; & i = n, \end{cases}$$

ahol

$$f_i(x) \overset{\text{det}}{=} (x_{i+1}-x)y_i+(x-x_i)y_{i+1} +$$

$$+ \frac{y_i'+\eta y_{i+1}'-(\eta+1)(y_{i+1}-y_i)}{\eta^2-1} \cdot \frac{\text{sh } p(x-x_i)-(x-x_i)\text{sh } p}{\text{sh } p - p}$$

$$- \frac{\eta y_i'+y_{i+1}'-(\eta+1)(y_{i+1}-y_i)}{\eta^2-1} \cdot \frac{\text{sh } p(x_{i+1}-x_i)-(x_{i+1}-x_i)\text{sh } p}{\text{sh } p - p}$$

ahol

$$\eta = \frac{\text{pchp} - \text{sh } p}{\text{sh } p - p} .$$

Képezve a második deriváltakat

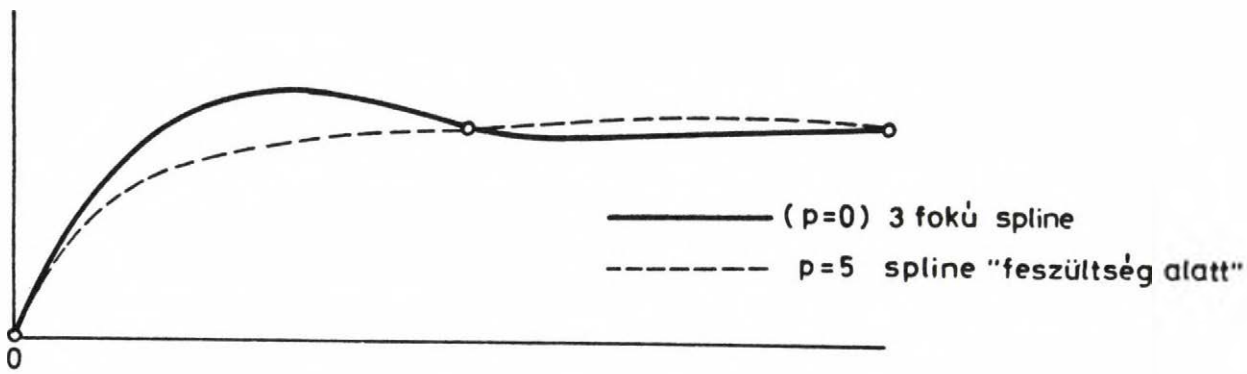
$$y_{i-1}' + 2\eta y_i' + y_{i+1}' = (\eta+1)(y_{i+1}-y_{i-1}),$$

ami n egyenletet ad az (n+2) darab y_i' ismeretlenre. Ha a végpontbeli érintők meredeksége, t.i. y_o' és y_{n+1}' ismert, máris csak n darab ismeretlen van. Ha viszont a végpontokbeli második derivált ismert a derivált helyett, akkor az alábbi két egyenlet még felirható:

$$2y_o'+y_1' = -y_o''\frac{\text{sh } p - p}{p^2 \text{ sh } p}(\eta^2-1) + (\eta+1)(y_1-y_o)$$

$$y_n'+\eta y_{n+1}' = y_{n+1}''\frac{\text{sh } p - p}{p^2 \text{ sh } p}(\eta^2-1) + (\eta+1)(y_{n+1}-y_n) \quad .$$

Megjegyezzük, hogy az egyenletrendszer mátrixában a diagonális elemek dominálnak, igy a mátrix invertálható, az egyenletrendszer egyértelmüen megoldható.



8. ábra

Hiba korrigálás spline "feszültség alatt" módszerrel

A szabadkézi rajzhoz való közelités érvényesül az alábbi két lokális módszerben /lsd Ackland [2o], Karup [19], Akima [14]/: Legyen adott egy függvénygörbe néhány pontjával. Ezt szakaszonként harmadfoku polinommal kivánjuk megközeliteni. A simaságot ugy garantálhatjuk, hogy minden pontban a szomszédos pontok ismeretében mi magunk számoljuk ki az érintő meredekségét. Az első, "oszkuláló" módszer szerint, ha adott a görbe $P_i(x_i,y_i)$ $i = 1,2,3$ pontja, akkor a /középső/ P_2-beli érintő t meredekségét a

$$t = (x_2-x_1)^2(y_3-y_2) + (x_3-x_2)^2(y_2-y_1)]/[(x_2-x_1)^2(x_3-x_2) +$$

$$(x_3-x_2)^2(x_2-x_1)] = [(x_3-x_2)m_1+(x_2-x_1)m_2]/[x_3-x_2)+(x_2-x_1)]$$

képlettel becsüljük.
A második /Akima-féle módszer/ öt pontot használ az érintő meghatározására.
Ha $m_1 = P_1P_2$ meredeksége

$m_2 = P_2P_3$ "

$m_3 = P_3P_4$ "

$m_4 = P_4P_5$ "

és a P_3 pontbeli érintőt akarjuk becsülni, akkor

$$t = (|m_4-m_3|m_2 +|m_2-m_1|m_3)/(|m_4-m_3|+|m_2-m_1|).$$

Ha az érintők már megvannak, akkor az egyes szakaszokra a Hermite-interpolációval harmadfoku görbéket illesztünk, amelyek az érintők számolási módszere miatt C^1-simaságu interpolációs görbét adnak. A végpontokban az érintőket a következőképp számolhatjuk: /persze az is megtehető, hogy a végpontokban

megadjuk az érintőket!/

$x_5-x_3 = x_4-x_2 = x_3-x_1$ és

$$(y_5-y_4)/(x_5-x_4) - (y_4-y_3)/(x_4-x_3) = (y_4-y_3)/(x_4-x_3) - (y_3-y_2)/(x_3-x_2) =$$

$$= (y_3-y_2)/(x_3-x_2) - (y_2-y_1)/(x_2-x_1) .$$

Az érintő meredekségét ezután az előző képlettel számolharjuk.

5.5. ÖSSZEFOGLALÓ MEGJEGYZÉSEK

Az eddig elmondottakból ugy tünik, hogy az "alakhü" betürajzolásban a görberajzoló és a tartománykitöltő algoritmusok szerepe némileg kisebb, mint az approximációs és interpolációs módszereké. Pontosabban ugy látszik, hogy a tartománykitöltés lényegében technikai probléma, s a görberajzolás is alá van rendelve az approximációnak /ha t.i. Bernstein-polinomokkal közelitünk, akkor a görberajzolást ezek jó rajzolására kell kihegyeznünk./ Az approximációs módszerek közül további kisérletezéssel választjuk ki a megfelelőt. Az előzőekben elmondottak alapján kedvező grafikai tulajdonságaikat figyelembevéve az alábbi módszerekkel érdemes kisérleteznünk:

- B-spline-okkal való közelités
- Bézier módszere /Bernstein-polinomokkal ill. B-spline-okkal/
- Coons féle approximáció
- Pavlidis kupszeletdarabokkal közelitő /Bézier jellegü/ módszere
- az "oszkuláló" közelités módszere
- Akima lokális módszere
- feszültség alatti spline-ok.

Mivel ez még igy is igen tekintélyes lista, igy azokat a módszereket részesitjük előnyben, amelyek viszonylag gyorsan számolhatóak, azaz

- Coons approximáció
- "oszkuláló" közelités
- Akima módszere
- Pavlidis módszere.

Ezek a megállapitások persze nem azt jelentik, hogy akár a fel nem sorolt módszereket is eleve el kellene vetnünk, de az utóbbi módszereket várhatóan nem tul nehéz implementálni, igy még az esetleges negativ választ is hamarabb értékelhetjük. /Megjegyezzük, hogy a Coons-approximáció alkalmazása tünik legegyszerübbnek, ennek grafikai tulajdonságai ugyan elég jók, de a többi felsorolt módszernél rosszabbak./ Ha az utolsó négy módszer valamelyike mellett döntünk, akkor a várható tárkapacitás remélhetőleg nem lesz a bevezető fejezetben szereplő becslés kétszeresénél rosszabb. Mivel ezek a módszerek a szabadkézi rajzhoz hasonlatosak, meglehetősen sok segédpont kell a ceruza irányitásához. Ha igy megengedhetetlenül nagy helyigény lépne fel, akkor előtérbe kerül a Bézier-módszer és a B-spline-okkal való közelités, amelyek kevesebb osztópontot használnak. Ezeknek a módszereknek a dekódolásával kapcsolatban felmerül az az ötlet, hogy a polinomok együtthatóit tároljuk. Ekkor azonban veszithetünk a tárkapacitásból, hiszen nem tudjuk, hogy a polinom /egyébiránt racionális/ együtthatói hány bitben ábrázolhatók.

Az implementálási sebesség szempontjából az alábbi kiértékelés tünik helyesnek:

I. Coons-módszer

II. Lokális módszerek /"oszkuláló", Akima/

III. Pavlidis módszer

IV. Bézier-módszer /Bernstein-polinomokkal és B-spline-okkal/

V. Magasabb foku B-spline-okkal ill. feszültség alatti spline-okkal való közelités.

Remélhetőleg a II. alatti módszerek már kielégitő megoldást fognak jelenteni. A III. módszer nagy előnye, hogy vele a "betütervezés" automatizálható. A IV. módszernél a Bézier-poligon kijelölésével és módositásával kapcsolatos, az V. módszernél elsősorban számolástechnikai problémákat kell megoldanunk. A helyigény minimalizálás szempontjából az 5.2. alatti /karaktergenerátorban használt!/ elliptikus ivekkel való közelités megerősiti azt a nézetünket, hogy az approximációs módszerek, mind a tárkapacitás, mind a dekódolás sebessége szempontjából megfelelő eszközt nyujtanak.

6. A GÖRBÉK MEGJELENÍTÉSE

Az $y = f(x)$ zárt képlettel megadott függvények megjelenitése általában egyenlőtlen rajzolatu görbét eredményez. Ezen a problémán segithetünk, ha a kirajzolandó görbét paraméteresen adjuk meg. A másik lehetőség az ábra javitására, ha két-két számitott függvénypont között egyenes szakaszokkal interpolálunk. A számitógépes grafikában ez a gyakoribb megoldás. Ezért kell foglalkoznunk az egyenes rajzolásával.

Egyenes rajzolása:

A mereven elhelyezkedő képpontok általában nem esnek egybe az ábrázolandó pont elméleti helyével, hanem annak közelében, legfeljebb egy raszternyi távolságban vannak. Bár az ábrázolás hibája igy nem nőhet egy rasztertávolság fölé, a rosszul elhelyezett pontok hullámzó, vagy változó vastagságu vonalat eredményeznek.

Az ismert egyenesrajzoló algoritmusok közül most két u.n. növekményes módszerrel müködőt ismertetünk.

Bresenham algoritmus:

Az algoritmus lényege, hogy igyekszik mindig csak egy koordináta irányában mozogni. Tehát mindaddig az X tengellyel párhuzamosan rajzolja ki a képpontokat, amig az elméleti egyenestől való elmaradása /távolsága/ egy az egyenes meredekségétől függő korlátot el nem ér. Ekkor az uj képpont koordinátái, az utolsó kirajzolthoz képest x-ben és y-ban is eltérnek egy-egy raszternyit.

Egyszerü szimmetrikus algoritmus:

Ennek az interpolációnak a lényege, hogy az i+1-ediknek kirajzolt képpont rendezői az i-edikhez képest

$$x_{i+1} = \text{trunc}(x_i + \Delta x)$$

$$y_{i+1} = \text{trunc}(y_i + \Delta y)$$

ahol a turnc függvény a képpontokra vett egészrész függvény, és

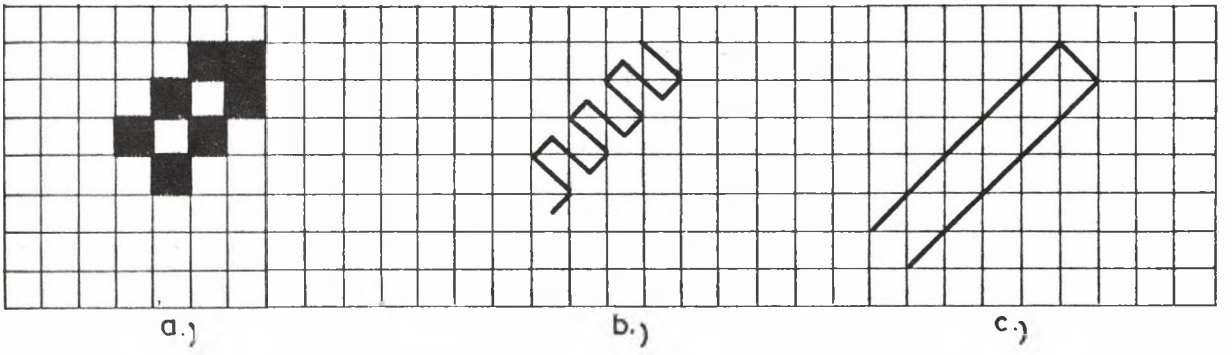
$$\Delta x = \frac{x1-x2}{\max[|x1-x2|;|y1-y2|]}$$

$$\Delta y = \frac{y1-y2}{\max[|x1-x2|;|y1-y2|]}$$

(x1,y1) és (x2,y2) az egyenes szakasz végpontjainak rendezői. Mindezek alapján általánosan görbevonalat ugy rajzolhatunk, hogy ha (x,y') a görbe utoljára kirajzolt pontjának képkoordinátái, és a következő számitott pont koordinátái (x+Δx,y), akkor, ha abs(y'-y) < 2Δy kirajzoljuk az (x+Δx,trunc(y)) pontot, különben egyenest rajzolunk az (x,y') és az (x+Δx,trunc(y)) pontok között.

7. KITÖLTŐ ALGORITMUSOK

Egy körvonalaival ábrázolt kép kiszinezésének alapkérdése, hogy egyértelmüen el kell dönteni, mi van a határvonalon belül. Nehézséget okoz a raszterpontos ábrázolás. A 9. ábráról nehéz eldönteni, hogy az (a) kép (b)-nek vagy (c)-nek a raszterképe-e, azaz kitöltendő-e vagy sem.



9. ábra

Egy raszterkép lehetséges vektoros értelmezése.

A [34]-ben áttekkintést találhatunk az alapvető kitöltő algoritmusokról.
Nézzünk néhány egyszerü stratégiát:
Pásztázzuk a képet az egyik, pl. az x-tengellyel párhuzamos sávokban. Hagyjuk meg a képpontok alapszinét mindaddig, amig e g y konturvonalba bele nem m e t s z ü n k.
Innentől kezdve fessük be a képpontokat mindaddig, amig ujra nem metszünk egy konturvonalat, és ekkor hagyjuk abba a festést. Ezt ismételjük az egész képen végig.

Az eljárás nehézsége, hogy a raszteres ábrázolás nem tökéletesen simul az elméleti görbealakra. Ezért nehéz annak eldöntése, hogy egy adott helyen metszettünk vagy csak érintet-

tünk egy görbét. Továb bonyolitja a helyzetet, ha fel kell ismernünk egy pontról, hogy kettős pont. Ezt pusztán a bittérkép alapján lehetetlen megmondani.

Heurisztikusabb gondolatmenettel dolgozik a következő algoritmus:

Keressünk a képen egy "belső" pontot.
Fessük be.
Rekurziven fessük be minden befestett pont szomszédját mindaddig, amig konturpontot nem érünk.
Folytassuk az eljárást mindaddig, amig festetlen belső pontot találunk. Az evvel kapcsolatos nehézség az, hogy több független, vagy egymást legfeljebb érintő zárt tartományt tartalmazó kép esetén hosszadalmas amig megbizonyosodunk,hogy minden tartományt megtaláltunk, másrészt meg kell különböztetnünk a kiszinezett belsőpontokat a konturvonal szinétől.

T. Pavlidis [24] munkájában több algoritmust is ajánl. Egyesek közülük a pásztázó algoritmust javitják azáltal, hogy minden pontnak a szomszédait is megvizsgálják abból a szempontból, hogy konturvonalak-e. Igy a végleges szinezés elött minden pontról háromszor dönt.
Pavlidis algoritmusainak másik csoportja, két menetben festi ki az ábrát. Először megcimkézi a konturvonalakat, - jobboldali, baloldali stb. - majd egy pásztázó algoritmussal kiértékeli a cimkéket.
Az ismertetett stratégiák alapvetően konvex sikidomok kitöltésére irányulnak. Esetünkben a kitöltendő felületek gyakran konkáv, nem egyszeresen összefüggő /"gyürüszerü"/ tartományok. Meg kell jegyeznünk, hogy jelen esetben a feladatból adódóan néhány feltételt ismerünk, amelyek segitenek a kitöltéshez algoritmust találni. Ezek a feltételek: az ábránk mindig fehér sávval kezdődik. Másrészt a betüknek nincs csak kép pont széles "vonaldarabja".

Érdemes néhány szót szólni arról, hogyan töltjük ki a felületeket. Sötét felületnél természetesen minden pontot befestünk. Szürke felületet nyerünk, ha egyeseket kihagyunk, azonban sávos vagy pettyes hatást érünk el, ha szisztematikusan választjuk meg a kihagyandó pontokat.

8. AZ ELKÉSZÜLT RENDSZER

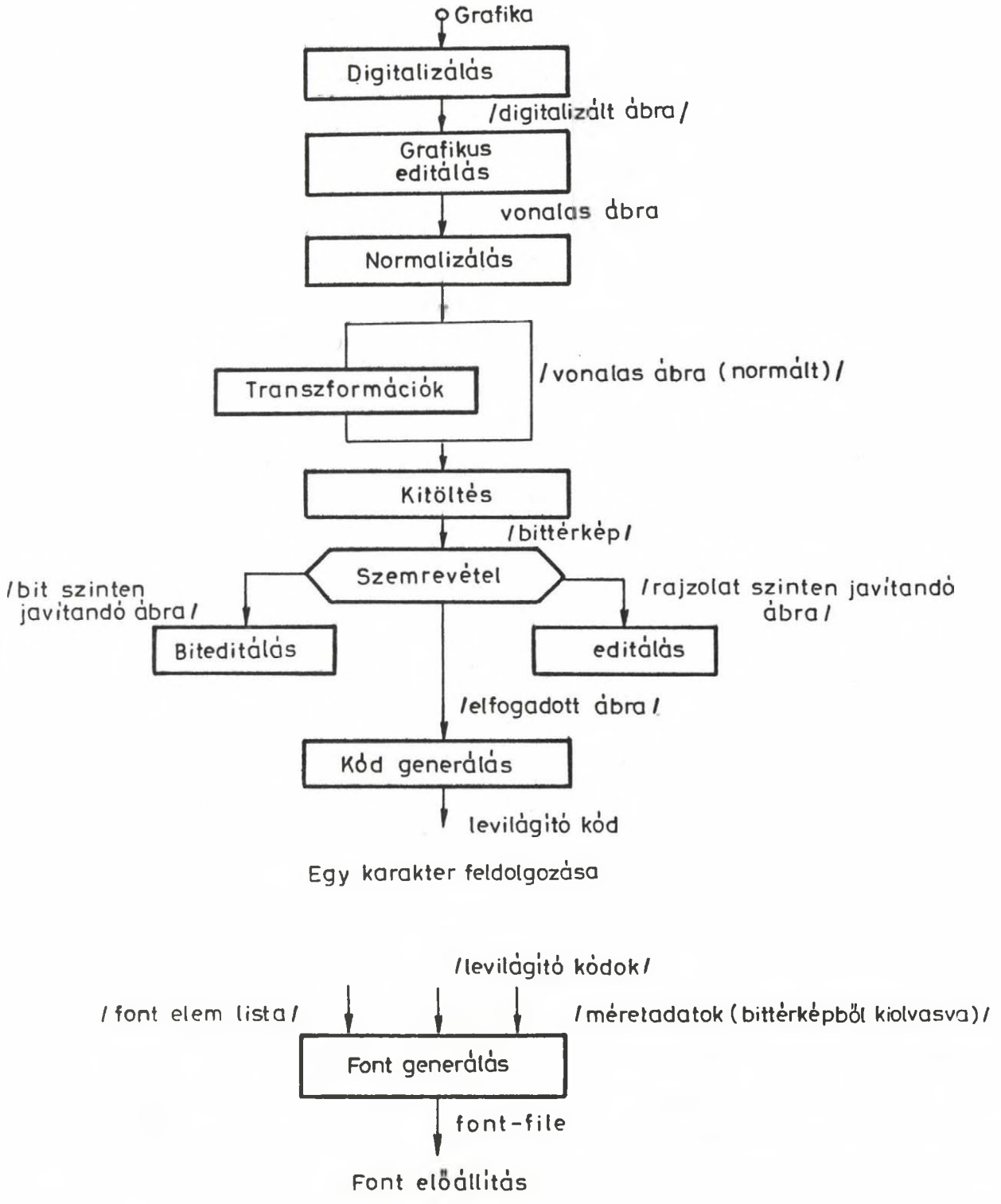
Végül talán érdemes ismertetni, hogy az előbbi megfontolások után milyen karakterfeldolgozó rendszert készitettünk el. A feldolgozás teljes folyamata a 10. ábrán látható.

Néhány lépés rövid ismertetése:

A grafikus editálás az a munkafázis, amellyel a karakterek konturvonalát előállitjuk. Gyakorlati szempontok miatt először a karaktereknek egy ugynevezett nyers vonalas ábráját állitjuk elő. Ennek jellemzője, hogy kismértékben eltér a mérete a kivánt névleges mérettől, másrészt nem tartalmaz minden olyan méretadatot, amelyek a karakternek nyomtatáskor a körülötte lévő karakterekhez képesti pontos helyét biztositja. /Pontosabban az egymás melletti betük távolsága megadott, de a sor alapvonala betünként más helyen lesz./
A fenti hiányokat pótolja a normalizálás, mégpedig egy font minden karakterére egységesen.
A transzformációk körébe tartozik a megkivánható méret és formaváltozatok előállitása. A formaváltozatok közül elkészült a dölt betük előállitására alkalmas program és a kövér illetve félkövér betük előállitására szolgáló.

Meg kell jegyezni, hogy a dölt betü nem azonos az ismert cursivval. Ugyanakkor az automatikusan előállitott kövér és félkövér karaktereket a grafikusnak szemrevételeznie és esetleg korrigálnia kell.
A kitöltött betük bittérképét szintén célszerü szemrevételezni. Részint, hogy meggyőződjön a kezelő, hogy sikerült a kitöltés, részint, hogy az apróbb szépséghibákat egy-egy bit ki-, illetve bevételével javitja.
A kód generálás feladata, hogy a leviláqitó berendezés/ek/ táplálására alkalmas, tömöritett kódot állitson elő.

A transzformációk kidolgozása kapcsán felmerült a gondolat, hogy reklám célra vagy egy nyomat szinesitésére transzformációval mesterségesen "torzitott" uj betüt hozunk létre. Két



10. ábra

A karakterfeldolgozó rendszer folyamat ábrája

ilyen kisérletet tettünk. Az egyikkel a betük körvonalait vastagitjuk, mig a belsejük üres marad. A másikkal tetszőleges arányban növelhetjük a függőleges és a vizszintes méreteket egymáshoz képest. Természetesen a két módszer kombinálható.

8.1. FONTOK ELŐÁLLITÁSA

A nyomdaiparban fontnak az azonos tipusu és méretü karakterek készletét nevezik. A fontok a nagy és kisbetükön, számokon, irásjeleken kivül más jeleket vagy szimbólumokat is tartalmazhatnak. Egy fontban 110-120 karakter van.
A számitógépes fényszedő rendszer egy-egy fontja legfeljebb 128 karaktert tartalmaz. Ahhoz, hogy a rendszer müködni tudjon, egyazon fontról több leiró file is készül. Az egyike a szedő program számára, ez csak méretadatokat tartalmaz, azért, hogy a program a karakterek helyét számitani tudja. Külön leiró file készül a levilágitó berendezés számára, amelyet a berendezés vezérlésként értelmez, hogy a karakter foltját levilágitsa. Érdemes észrevenni, hogy a hagyományos, ólombetüs esetben a betübélyeg egymaga hordozta a méret és alaki információt. Adatszerkezetét tekintve a font leirásának mindkét alakja azonos: egy katalógus blokk után a karakterek megfelelő leirásának felsorolása következik.

8.2. A GRAFIKUS EDITOR

A nyomdai betük előkészitését segitő programrendszer központi eleme a grafikus editor. Ennek a programnak a segitségével lehet a karaktereket körberajzolni, a rajzolatot javitani, vagy a bittérképet módositani. Természetesen szabad rajzolásra is alkalmazható. A program alapgondolata a következő: három átlátszó rajzlapot terit a felhasználó elé, és minden rajzlaphoz hozzácsatol egy rajzeszköz készletet, amely csakis azon a lapon használható. A kezelő a kezdeti üzemmódválasztással dönti el, hogy mit tölt, melyik rajzlapra és milyen rajzeszközt választ hozzá.

A legalsó rajzlapra valamilyen bittérképet tölthet, vagy üresen hagyja. A bittérkép lehet a digitalizált kép, vagy a kitöltő által kibocsátott bittérkép. A középső lapot vagy üresen hagyja, vagy egy már létező vonalas ábrát tölt rá. Végül, a legfelső lapot vagy üresen hagyja, vagy egy létező keretlapot tölt rá.
A legalsó rajzlapon, ha a bittérkép a kitöltő programtól ered, képpontokat lehet törölni, vagy az ábrához illeszteni.
A legfelső rajzlapon a rajzolat koordináta rendszerével párhuzamos egyenesekkel a karakterek jellegzetes méretvonalai jelölhetők ki.

Általános érdeklődésre a középső rajzlap rajzeszközei tarthatnak számot. Amit itt találunk, az egy általános célu, szabadkézi rajzolást támogató grafikus program.
A felhasználó a rajzlapon rajzelemek és müveletek segitségével dolgozhat.

A rajzelemek segitségével pont, egyenes és görbék rajzolhatók. Az utóbbiak lehetnek körök, ellipszisek, vagy egészen általános görbe vonalak, amelyeket spline-okkal közelitünk.
A rajzelemek között két "radir" is található. Egyrészt kitörölhető egy vonalszakasz, amely több, folytonosan rajzolt rajzelemet is magábafoglalhat. A törlés határainak nem kell rajzelem végpontra esni. Ez a törlés az eredményben is szerepel. Másrészt kitörölhetők a képen lévő, az addigi szerkesztésből eredő segédvonalak. Ez a törlés természetesen nem okoz a végeredményben nyomot.
A rajzelemeket néhány pontjukkal határozzuk meg. Pontositva; egyenest végpontjaival, görbét végpontjaival, a végpontokban rajzolt érintővel és néhány belső pontjával lehet megadni, mig szakasz törléshez a törlendő szakasz végpontjai és egy közbenső pontja szükségesek.
A müveletek mindig egy vagy több, teljes rajzelemből összefüzött listára vonatkoznak. A listába füzéshez a program felkinálja egymás után a rajzolat összes elemét, és a kezelő választja ki közülük azokat, amelyekkel müveleteket akar végezni.

A végezhető müveletek, eltolás, tükrözés-eltolás, /a tükrözés pontra, vagy bármilyen irányu tengelyre/ elem törlés és makrozás. Makrozás alatt azt értjük, hogy a kezelő néhány képelemből készletrajzot másol ki, ezt háttér memóriába teheti, vagy egy már a háttér memóriába tett készletet elővehet, és a kép egy megjelölt részére másolhatja, hozzáilleszthetí.

Talán nem érdektelen megemliteni, hogy milyen tapasztalatok alapján és hogyan döntöttünk a görberajzolás kérdésében:

megvalósitottuk az "oszkuláló", "Akima" és "Coons-féle approximációt mind paraméteresen adott, mind függvénnyel való közelitésekre. A függvénnyel való közelités elleni fő érv az volt, hogy egyrészt nagyon megköti a rajzoló kezét, másrészt a függőleges érintők kezelésére /és betüknél ez tipikus!/ nem sikerült jól müködő konvenciót találni. Várakozásunkkal némileg ellentétben egyik lokális módszer /sem az oszkuláló, sem az Akima-féle/ "sem birta ki" a paraméteres görbeábrázolást. Mindkettőről elmondható, hogy zavaróan sok inflexiós pont keletkezett a rajzon. A Coons-approximáció fennmaradt a rostán és bár a várakozásnak megfelelően itt is /különösen nagyitás után/ keletkeznek nem kivánt inflexiók, számuk jóval kevesebb, mint a a "lokális" módszereknél. /Ráadásul a nagyitás során keletkező inflexiók jórészéért nem a Coons módszer, hanem a közbülső pontok tul sürü felvétele a bünös./ A Coons-módszer viszonylag jó grafikai és számolási tulajdonságai bőven ellensulyozták a fölösleges inflexiók létrejöttét. Egészen pontosan mi a 5.3. pontban paraméteres megadásu görbékre részletesen leirt módon használjuk a Coons-approximációt.

Ezen kivül az elkészitett grafikus editorral lehet speciális másodfoku görbéket /kör és ellipszisivet/ rajzolni és transzformálni.
Kört rekurzive az

$$x_{n+1} = u + (x_n - u)\cos\Theta + (y_n - v)\sin\Theta$$

$$y_{n+1} = v + (y_n - v)\cos\Theta - (x_n - u)\sin\Theta$$

képletekkel /lásd pl. [21]/ rajzolunk, mig a többi másodfoku rajzelemet kör /lineáris/ transzformációjával állitjuk elő. /Megemlitjük, hogy a betütervezésben a kör és ellipszisivek szerepe korlátozott, főleg cimbetüknél fordulnak elő./

I R O D A L O M J E G Y Z É K

[1] WALTER, G.D.: Typesetting, Scientific American, 220/no.5, 1969, 60-69.

[2] GORDON, W.J. & RIESENFELD, F.R.: Bernstein-Bézier methods for the computeraided design of freeform curves and surfaces, GMR-1176, 1972

[3] GORDON, W.J. & RIESENFELD, F.R.: B-spline curves and surfaces, in: R.E. Barnhill & Riesenfeld eds.: Computer aided geometric design, 1974

[4] GILOI, W.: Interactive computer graphics, Prentice-Hall, Englewood Cliffs, 1978.

[5] DE BOOR, C.: On calculating with B-splines, Journal of Approx. Theory 6/1 1972 July.

[6] SAMET, H.: Region representation: Quadtrees from boundary codes, Comm. ACM 1980, 163-170.

[7] SAMET, H.: Region representation: Quadtrees from binary arrays, Comp. Gr. Image Proc. 13(1980), 88-93.

[8] SAMET, H.: An algorithm for converting rasters to quadtrees, IEEE Trans. Pattern Anal. Mach. Intell. 3(1981), 93-95.

[9] DYER, C.R. & ROSENFELD, A. & SAMET, H.: Region representation: Boundary codes from quadtrees, Comm. ACM March 1980, 171-179.

[10] HUNTER, G.M. & STEIGLITZ, K.: Linear transformation of pictures represented by quadtrees, Comp. Gr. & Image Proc. 10(1979), 189-296.

[11] HUNTER, G.M. & STEIGLITZ, K.: Operations on image using quadtrees, IEEE Trans. on Pattern Anal. & March. Intell. PAMI-1 (1979), 145-153.

[12] COONS, S.A.: Modification of the shape of piecewise curves, Computer Aided Design 9 (3) 178-180, (1977)

[13] GHOSH, P.K. & MUDUR, S.P.: Parametric curves for graphic design systems, The Comp. Journal 26 No4. (1983), 312-319.

[14] AKIMA, H.: A new method of interpolation and smooth curve fitting based on local procedures, Journal of ACM, 17 No.4. (1970), 589-602.

[15] PAVLIDIS, T.: Curve fitting with conic splines, ACM Trans on Graphics, 2 No1. (1983), 1-31.

[16] STRASSER, W.: Fast curve and surfaceggeneration for interactive shape design, Computers in Industry 3(1982), 105-111.

[17] TERVONEN, M. & HAKALAHTI, H. & LAPPALAINEN, P.: A microprogrammable character generator for a CRT phototypesetting system, in: M. Sami L. Thompson, L. Mezzalira (eds.): Microprocessor systems, North-Holland, 1980.

[18] SCHWEIKERT, D.G.: An interpolation curve using a spline in tension, J. Math and Phys. 45 (1966), 312-317.

[19] KARUP, J.: On a new mechanical method of graduation, in: Transactions of the second Int. Actuarial Congr., C.&E. Layton, London, 1899, 78-109.

[2o] ACKLAND, T.G.: ON osculatory interpolation where the given values of the function are at unequal intervals, J. Inst. Actuar. 49(1915) 369-375.

[21] NEWMAN, W.M. & SPROULL, R.F.: Principles of Interactiv Computer Graphics. McGraw-Hill.

[22] LUCAS, M.: La realisation des logiciels graphiques interactifs. Eyrolles. 1982.

[23] FOLEY, J.D. & VAN DAM A.: Fundamentals of Interactive Computer Graphics. Addison-Wesley 1982.

[24] PAVLIDIS, T.: Filling Algorithms for Raster Graphics. Comp. Graph. and Image Proc. 10, 126-141 (1979).

[25] OLIVER, M.A. & WIESMAN, N.E.: Operations on Quadtree Leaves and Related Image Areas. The Comp. Journ. vol.26. no.4. 1983.

[26] SPROULL, R.F.: Using Program Transformations to Derive Line-Drawing Algorithms. acm. Trans. on graph. vol.1. no.4. 1982. oct.

[27] KNUTH, D.E.: Tex and Metafont, Digital Press, 1979.

[28] SZANTO, T.: A betü, Gondolat, Budapest, 1972.

[29] LAFATA, P. & ROSEN, J.B.: An Interactive Display for Approximation of Linear Programming, CACM 13, 11 Nov. 1970 pp. 651-659.

[3o] DE BOOR, C.: A Practical Guide to Splines, Springer, New York, 1978.

[31] PAVLIDIS, T.: Structural Pattern Recognition, Springer, New York, 1977.

[32] SZIDAROVSZKY, F.: Bevezetés a numerikus módszerekbe, Közg. és Jogi Kiadó, Budapest, 1974.

[33] HENRICI, P.: Numerikus analizis, Müszaki Kiadó, Budapest, 1986.

[34] REVICZKY, J.: A számitógépes grafika területkitöltõ algoritmusai, MTA SZTAKI Tanulmányok, 172/1985

A TANULMÁNYOK SOROZATBAN 1986-BAN MEGJELENTEK:

180/1986 K.N. Cimev: Separable sets of arguments of functions

181/1986 Renner Gábor: Kör approximációja a számitógépes geometriai tervezésben

182/1986 Proceedings of the Joint Bulgarian-Hungarian Workshop on "Mathematical Cybernetics and Data Processing" Vol I

183/1986 Proceedings of the Joint Bulgarian-Hungarian Workshop on "Mathematical Cybernetics and Data Processing" Vol II

184/1986 Ho Thuan: Contribution to the theory of relational databases

185/1986 Proceedings of the 4th International Meeting of Young Computer Scientists, IMICS'86 /Smolenice, 1986/ Edited by: J. Demetrovics and J. Kelemen

186/1986 PUBLIKÁCIOK - PUBLICATIONS 1985 Szerkesztette: Petróczy Judit

187/1986 Proceedings of the winter school on conceptual modelling /Visegrád, 27-30 January, 1986/ Editors: E. Knuth, A. Márkus

188/1986 Lengyel Tamás: A Cluster analizis néhány kombinatorikai és valószinüségszámitási problémája

189/1986 Bernus Péter: Gyártórendszerek funkcionális analizise és szintézise

190/1986 Hernádi Ágnes: A tipus fogalma és szerepe a modellezésben

191/1986 Vu Duc Thi: Funkcionális függőséggel kapcsolatos néhány kombinatorikai jellegü vizsgálat a relációs adatmodellben

192/1986 Márkusz Zsuzsanna: P a p e r s on Many-sorted logic as a tool for modelling

193/1986 KNVVT Conference on Automation of Information Processing on Personal Computers Budapest, May 5-9, 1986 Vol I. Editor: I. Ratkó

194/1986 KNVVT Conference on Automation of Information Processing on Personal Computers Budapest, May 5-9, 1986 Vol II. Editor: I. Ratkó

1987-BEN EDDIG MEGJELENTEK:

195/1987 Telegdi László: Bináris változók strukturájának vizsgálata

196/1987 Rónyai Lajos: Algebrai algoritmusok

197/1987 Hernádi Ágnes - Bodó Zoltán - Knuth Előd: A tudásábrázolás technikái és gépi eszközei

198/1987 Miguel Fonfria Atan: A data base management system developed for the Cuban minicomputer CID 300/10

199/1987 Bach Iván - Farkas Ernő - Naszódi Mátyás: A magyar nyelv elemzése számitógéppel

200/1987 Publikációk'86 - Publications'86 Szerkesztette: Petróczy Judit