

Heiko Unt

## **Digitaalne tüpograafia**

Tartu Kõrgem Kunstikool

Arvutigraafika osakond  
Meediakunsti eriala

Heiko Unt

# **Digitaalne tüpograafia**

Diplomitöö

Juhendaja Andres Rõhu

Tartu  
2002

## **EESSÕNA**

Peame ennast infotehnoloogia ja arvutikasutajate hulga poolest edukaks rahvaks - väikeses Eestis on sadu tuhandeid arvutikasutajaid. Astusime arvutiseerumise rongile 90-ndate alguses. Sellest ajast on palju muutunud ja meie huvi digitaalsete töö võimaluste, kommunikatsioonide, kunsti jms vastu ei ole sugugi jähnenud. Emor teatab oma uuringus: "Eesti firmad on infotehnoloogiliselt oma Läti ja Leedu konkurentidest sammu võrra ees. Kui Eesti firmadest rakendab oma äritegevuses arvutit kolm neljandikku, siis Leedus on arvutiostuni jõudnud kaks kolmandikku ja Lätis pooled firmad." (*Eesti on Baltikumi IT valdkonna liider*, 23.05.2002 - AS Emor - [www.emor.ee](http://www.emor.ee))

90-ndate alguses oli arvutid e riistvara kallis ja ta võimalused paljudes valdkondades piiratud. Aastatega harjusime hindadega ja arvuti muutus aina populaarsemaks. Tekkisid eelistused ja konkurents erineva riistvara vahel, kujunesid välja eestimaised Macintosh arvutite fanaatikud ning PC kasutajad jagunesid AMD ja Inteli eelistajateks. Kui keegi meist peaks soovima soetada uut arvutit, võib ta külastada rohkeid eestimaiseid internetilehekülgi, kus võrreldakse ja soovitatakse erinevaid tehnilisi lahendusi ning on võimalik hankida infot hindade ja edasimüüjate kohta.

Pelgalt arvutist (riistvarast) ei piisa, et muuta teda meile kasulikuks ja asendamatuks kaaslaseks, vaja on mitmesugust tarkvara, alustades operatsioonisüsteemist ja lõpetades pisemate rakendustega nagu CD-pleier. Paljud ei ole veel teadvustanud endale, et iga vähegi mahukam tööprotsess eeldab, et tarkvarale kulub vähemalt samas suurusjärgus raha kui riistvarale, tihti isegi rohkem. Siin on meie kasutajaskond aga lapsekingades, meil on saadaval küll enamus tarkvara (vaatamata kuudepikkustele tarnetähtaegadele), aga puudu jääb tihti kogunud kasutajaskonnast (miks mitte ka koolitatud edasimüüjatest), kes jagaks kogemusi, adekvaatseid hinnanguid, rakendustel põhinevaid võrdlusi jne. Iga arvutikasutaja puutub pidevalt kokku arvutišriftidega, mis on samuti pisikesed programmid. Šriftid teostavad keerukaid protsesse, parandamaks teksti loetavust ja tüpograafilisi omadusi. Vahel kuuleme, et räägitakse näiteks PostScript või True-Type šriftidest, enamasti piirdub see aga väga pealiskaudsete eelistuste või alusetute väidete-ga, mis on ajast ja arust.

Kust šriftid tulevad? Kas šrifte ei võigi lihtsalt internetist tõmmata või sõbralt laenata? Selle küsimuse vastust ei taha enamus eestlasi teada. BSA Eesti tunnistab šriftid küll autorikaitse objektiks, aga tõdeb samas, et esialgu ei ole esmaseks prioriteediks nende legaalsuse kontrollimine. Sellest ka meie üldine teadmatuse kogu šriftide tehnoloogiast ja põhierinevustest erinevate formaatide vahel. Kui ei maksa, pole ju oluline uurida, mis formaat oleks mõttekam või kas ma ikka kasutan kõiki rakendusi, mis ma oma raha eest olen saanud. Paraku, kui meil isegi on huvi, leidub antud teema kohta eesti-keelset materjali minimaalselt.

Loodan, et minu töö annab mõned pidepunktid avastamiseks piiratud digitaalsete šriftide ja nende tüpograafiliste omaduste maailma.

Heiko Unt  
Tartu Kõrgem Kunstikool

# SISUKORD

Eessõna.....	3
Sisukord.....	4
Sissejuhatus .....	5
<b>1. Šriftid, mis neil vahet on? .....</b>	<b>6</b>
Lühike ajalugu .....	6
Sarnasused .....	7
Tehnilised erinevused .....	8
Edasised praktilised erinevused .....	10
Multiple Masters .....	12
QuickDraw GX & GX šriftid .....	14
Unicode .....	15
Rahvusvaheline keelte toetus & WGL4 .....	15
OpenType .....	16
Mida toob tulevik? .....	17
Peatüki kokkuvõte .....	18
<b>2. True Type - šriftide ajalugu.....</b>	<b>19</b>
Kommentaariid Greg Hitchcock'ilt .....	21
<b>3. Times (New) Roman.....</b>	<b>22</b>
<b>4. Šrifti kontuurid .....</b>	<b>24</b>
BEZIER'i kõver .....	24
Vektorite matemaatika .....	25
Täispinnad paremale .....	26
<b>5. Vihjamine .....</b>	<b>27</b>
Vihjamine šriftis .....	28
TrueType ja automaatne vihjamine.....	30
<b>6. Šriftid ja "anti-aliasing" .....</b>	<b>31</b>
Mis on anti-aliasing? .....	31
Anti-aliasing ja tüpograafia .....	31
Anti-aliasing ja Macintosh .....	32
<b>7. OpenType šrift .....</b>	<b>33</b>
Pilguheit OpenType omadustele .....	33
TrueType vs OpenType .....	35
OpenType terminoloogia .....	35
<b>8. ClearType .....</b>	<b>40</b>
ClearType tehnoloogia .....	40
<b>9. Pikelšriftid .....</b>	<b>42</b>
<b>Kokkuvõte.....</b>	<b>43</b>
<b>Lisa 1. Mõisted ja terminid .....</b>	<b>44</b>
<b>Lisa 2. OpenType tabelid .....</b>	<b>45</b>

# SISSEJUHATUS

**Tüpograafia** tegeleb tekstiga – tähtedega, nende konstrueerimisega, paigutamisega, joondamisega jne. Läbi aegade on püütud muuta tekste kergemini loetavamaks ja esteetiliselt nauditavamaks. **Digitaalne tüpograafia** ei ole midagi uut, tegemist on samade vanade probleemidega - teksti loetavus, tähtede paigutuse rütm, tähed on segased või ebakorrektsed - ainult selle vahega, et tegemist on arvutitega ja segadus digitaalne. Tänapäeva arvutites tegelevad digitaalse tüpograafia probleemidega šrifid.

Olen seadnud töö eesmärgiks anda ülevaade suuremate šrifti-põlvkondade tekkest ja arengust tiheda konkurentsi tingimustes. Tahan, et lugeja saaks infot peamiste uuenduste ja erinevuste kohta, mis eristavad šrifte. Minu eesmärk on teadvustada, et šrift on väga keerukas ja komplitseeritud tarkvaraline lahendus, millele esitatakse sageli tohutuid nõudmisi. Materjal peaks aitama šriftikasutajatel mõista, milline konkreetne tarkvaraline lahendus oleks talle kõige mõttekam. Soovin, et lugejatel tekkiks suurem austus šriftiloojate töö vastu ja et nad aksepteeriksid šrifte kui disaini ning autorikaitse objekte.

Olenevalt šrifti formaadist ja otstarbest on aga tehnilised parameetrid pisut erinevad. Käesolev töö on võimaluste piires ülesehitatud võrdlustele ja vastandustele. Antud töö raames ei ürita ma anda täielikku ülevaadet ühestki konkreetsest tehnoloogiast, käsitlen levinumaid probleeme ja populaarsemaid uuendusi. Samuti vaatlen detailsemalt tehnilisi rakendusi, mida kasutavad kõik digitaalsed šrifid. Tööst võib leida rohkesti viiteid täpsemale infole.

Töö algab kahe suurema šriftipõlvkonna võrdlusega - PostScript vs TrueType, jätkan erinevate tehniliste lahenduste tutvustamisega, vaatlen Multiple Masters, QuickDraw GX, Unicode, WGL4 ja OpenType tehnoloogiaid. Käsitlen Times Roman šrifti kui ajaloolist mõõdupuud, defineerin ja seletan lahti mõisted nagu: šrifti kontuurid, Bezier'i kõver, vihjamine ja *anti-aliasing*. Annan ülevaate viimase šrifti-põlvkonna uuendustest/võimalustest - OpenType. Lõpetuseks peatun interneti- ja ekraanišriftidel.

Teema valikul sai otsustavaks isiklik huvi digitaalne tüpograafia vastu. Määrav oli ka see, et antud teemat on Eestis suhteliselt vähe ja pealiskaudselt käsitletud. Näen töös ka võimalust valmistada ette võimalikku digitaalse tüpograafia loengu pedagoogilist materjali.

Töö tugineb suuresti šrifti- ja tarkvaratootjate lehekülgedel olevale materjalile, olen kasutanud pressiteateid ja tarkvaraarendajatele mõeldud tehniliste lahenduste spetsifikatsioone. Samuti on töös kasutatud šriftide arenguprotsessis osalenud inimeste (*Sampo Kaasila - Macintosh, Greg Hitchcock - Microsoft, Norman Walsh - Sun, jt*) ja teemat uurinud teadusasutuste (*School of Computer Science at McGill University*) kommentaare.

## Šriftid, mis neil vahet on?

**Šrift** (ing *font*) - kogum sümboleid või tähemärke. Vanasti, kui trükkimiseks kasutati metalltähti, tähendas see tähestiku jagu tähti ja muid vajalike märke kindlal suurusel teksti ladumiseks. Digitaalse teksti maailmas on šrift ise tähemärkide kogum või digitaalne info selle konstrueerimiseks. **Tähemärk** (ing *glyph*) on kirjamärgi esitus, konkreetne üks märk, näiteks suurtäht "A", šriftis Times Roman. **Kirjamärk** (ing *character*) on arvuti klaviatuurilt sisestatud märk näiteks "a". Ühele kirjamärgile võib vastata mitu tähemärki, näiteks võib "a" olla ekraanil nii suur kui väike "a", ka šrift võib olla erinev näiteks Arial või Times. Selleks et leida/kuvada/printida tähemärki, kasutatakse šriftis olevaid rakendusi, mis väljastavad vastavalt kasutaja poolt sisestatud kirjamärgi koodile sellele vastava tähemärgi.

On kaks peamiselt kasutatavat tehnoloogiliselt erinevat šrifti-põlvkonda - **TrueType** (TT), mille on loonud Apple kompanii (süsteemipõhiselt toetatud 95% maailmas kasutatavatest arvutitest) ja **PostScript** (PS). PS šriftid on välja arendanud Adobe Systems, arvatakse, et maailmas on kasutusel rohkem kui 20 000 PS šrifti, genereeritud erinevate kompaniide ja üksikute disainerite poolt.

## Lühike ajalugu

PostScript edestab TrueType tehnoloogiat umbes kuue aastaga. Esmalt oli palju erinevaid digitaalseid formaate šriftide jaoks, kuid ükski neist polnud kujunenud juhtivaks standardiks. 1985 aastal adopteeris Apple Adobe PostScript keele (*page description language - PDL*) oma AppleLaserWriter printeritele. Ajastus ühildus graafika-tarkvara pealetungi algusega, vallandus revolutsioon tekstiküljenduses.

Varsti võeti PS keel kasutusele paljudes tipp-tasemel seadmetes, kus ta kujunes levinud opereerimismooduseks ja paljude graafikaprogrammide baaskeeleks. Ülemvõim tundus olevat kindlustatud, sellel hetkel oli Adobe'1 kontroll kogu PS tehnoloogia üle. Keele käsustik oli avalikult kättesaadav ning võimaldas konkurentidel luua PS'il põhinevat tarkvara, võistlemaks Adobe'ga, samas ei olnud võimalik tõlkida šriftidele hädavajalikku **vihjamist** (ing *hinting* - šrifti või rakendustarkvarasse programmeeritud kogumik infot, selleks et korrastada/parandada šrifti paigutust ja tähemärkide detaile). Seda sellepärast, et ei olnud avalikustatud PS šrifti **spetsifikatsioon** (ing *specification* - detailne funktsioonide kirjeldus, vajalik tarkvaraarendajatele ja šriftide disaineritele) **PS Type 1**'le, mis sisaldab vihjamist. Adobe avalikustas küll **PS Type 3** šrifti spetsifikatsiooni, millel olid mõned väiksemad eelised, kuid mis ei töötanud hästi madala ekraanilahutussagedusega seadmetel.

Suurtele tarkvara arendajatele (Apple, Microsoft, ja hiljem IBM) sai peagi selgeks, et vajalik oleks selline **skaleeritavate** (ing *scalable* - muudetava suurusega) **šriftide süsteem**, mida tarkvara toetaks juba operatsioonisüsteemi tasandilt. Aga ei Apple ega ka

---

*Mõisted on täiendavalt defineeritud töö lõpus.*

Microsoft tahtnud, et keegi väljastpoolt oma firmat kontrolliks mõnd võtmerakendust nende operatsioonisüsteemis. Seega aretas Apple oma isikliku šriftisüsteemi, algselt koodnimega Bass, hiljem Royal, ja lõpuks avalikusele tutvustatud kui TrueType. Apple jagas tehnoloogiat Microsoft'iga, vastutasuks pidi Microsoft looma PS keelel baseeruva graafikamootori (selleks sai TrueImage, mis oli paraku aga väga vigane ja ei jõudnudki kunagi Apple poolt kasutusse). TT šrifti spetsifikatsioon avalikustati ja nende rakendused ehitati sisse kõigile Mac'i ja Windows'i järgnenud operatsioonisüsteemidele tänaseni ja ilmselt ka veel edaspidi.

Adobe vasturünnak algas kaua varjatud PS1 šrifti spetsifikatsiooni avalikustamisega märtsis 1990. Sellele järgnes **Adobe Type Manager** (ATM) tarkvara tutvustamine 1990 keskel. ATM skaleerib Type 1 (aga mitte Type 3) PS šrifte ekraanil kuvamiseks ja edastamiseks PS toetusega ja toetuseta printeritele.

1991 aasta alguses oli TT tugi olemas Macintosh arvutitele, järgnes töötlus Windows 3.1'le. Nüüd said TT ja ATM'i kasutajad oma silmaga vaadelda ja võrrelda ekraanil erineva suurusega šrifte.

Aga aeg läheb ja piirjooned tehnoloogiate vahel muutuvad üha udusemaks. Täna on TT toetus sisse ehitatud paljudele PS Level 2 rakendustele ja PS3 puhul on see standard. Sarnaselt on ehitatud ATM'i rasteriseerimise tehnoloogia Windows NT 5.0 operatsioonisüsteemi sisse kõrvuti TT tehnoloogiaga. PS1 šriftid on süsteempõhiselt toetatud Windows 2000, Windows XP ja Mac OS X operatsioonisüsteemides, vanemad versioonid vajavad Adobe priivara (ing *freeware* - tasuta tarkvara) Adobe Type Manager Light.

Apple OS X operatsioonisüsteem kasutab ekraani PostScript'i, kui alust kogu graafika mudelile, mis annab PS šriftidele kaasasündinud süsteempõhise toetuse. Uues **OpenType** formaadis (vaadeldud täiendavalt edaspidi), võib nii TT kui PS kontuurid panna ühisesse formeeringusse.

## Sarnasused

TrueType ja PostScript on mõlemad multiplatvormilised ja kontuuršrifti standardid, šriftide spetsifikatsioonid on avalikustatud ja kõigile kättesaadavad. **Multiplatvorm** (ing *multi-platform*) tähendab, et šriftid on kasutatavad paljudel erinevatel arvutisüsteemidel. **Kontuuršrift** (ing *outline font*) tähendab, et šrifti kirjeldatakse tähe (tähemärgi) piirjoonte järgi, tähistades punkte, mida ühendavad sirged ja kõverad.

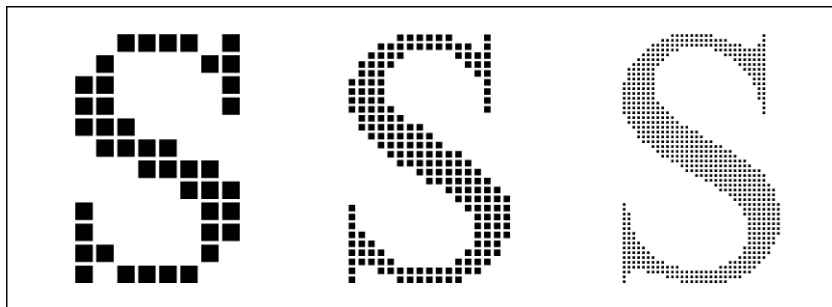
Kontuuršrifti tehnoloogia peidab lahendust juba iseeneses, arvestades et kontuurid, tänu oma vektorloomusele, on suurendatavad ja vähendatavad erinevatele suurustele, kaotamata seejuures kvaliteedis. Sõltuvalt kasutatavast tekstiõõtlustarkvarast ja operatsioonisüsteemist, võib šriftil siiski olla maksimaalne ja minimaalne lubatud suurus, aga enamusel kasutajatest ei olegi kunagi vajadust ega soovi neid piire ületada.

Väljundseadmetele, näiteks printerile saadetakse kontuuršrifti info punktidenä. Protsessi, mille käigus šrifti kontuur konverditakse **piksel- e rasterpinnaks** (ing *bitmap* - igale kujutise andmeelemendile/punktile vastab bitt), vastavalt väljundseadme vajadustele, nimetatakse **rasteriseerimiseks**.

Tavaliselt mõõdetakse teksti suurust **punktides** (ing *points*) – üks punkt on tüpograafiline mõõtühik e 1/72 tollist (ing *inch* ~ 2,4 cm), kogu ekraanigraafika toimibki vastavalt sellele e ekraani **resolutsioon** (punktide arv pikkusühiku kohta) on 72 ppi (*points per inch* - punkti tolli kohta). Väljaprinti puhul on samuti tegemist punktidega, siis juba aga trü-

kiseadme poolt võimaldatud punktide (ing *dot*) suuruse ja tihedusega, suhet väljendab ühik dpi (*dot per inch* - punkti tolli kohta).

Kui ei ole piisavalt punkte, et konstrueerida tähemärki (näiteks väga väikese kirja puhul või väga madala resolutsiooni korral), võib esineda ebajärjekindlust samade tähemärkide esitamisel defineeritud suurusel, tulenevalt sellest, kuidas tekstirida ühtib punktireaga väljundseadmes. Tüüpilisem viga on sellistel juhtudel, et tähepostide laius võib varieeruda, kuigi ta ei tohiks. Halvem on, kui tähemärkide põhitunnused kipuvad kaduma väikeste tekstide puhul.



Joonis 1. Rasteriseerimine erinevatel resolutsioonidel: 18 punktine kiri 72 dpi, 18 punktine kiri 144 dpi ja 18 punktine kiri 300 dpi resolutsiooniga.

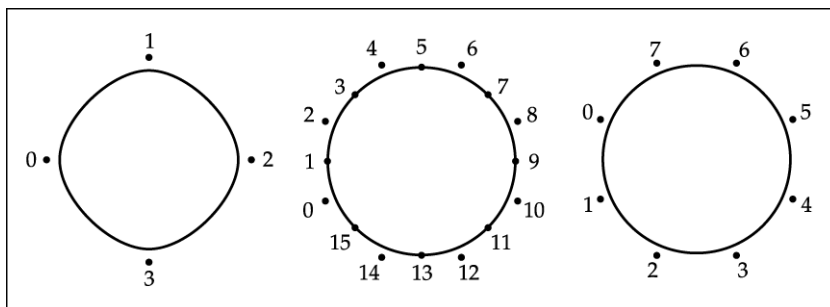
PS1 ja TT šrift, mõlemad üritavad sarnaselt vältida eelmainitud ebakorrektsusi, teostades protsessi vihjamine. Vihjamine on šrifti programmeeritud kogumik infot selleks, et korrastada ja parandada šrifti layouti ja tähemärkide detaile. Lisaks sellele, et vihjamine aitab vältida ebaühtlasi tähemärke väljatrüki puhul, on ta asendamatu ka multimeedia toodete ja interneti lahenduste juures, kus on väga oluline, et tekstid oleksid monitoril korralikult loetavad ja ei väsitaks liigselt kasutaja silmi.

## Tehnilised erinevused

Esimene tehniline erinevus TrueType ja PostScript šriftide vahel on see, et nad kasutavad erinevaid matemaatilisi valemeid, kirjeldamaks kõveraid šrifti piirjoontes. Seepärast on enamasti ebakorrektna ka konverteerimine ühest tehnoloogiast teise: matemaatika terminitega rääkides, ruutvõrrandil põhinevad B-kõverad TT šriftis on osa kuupvõrrandil põhinevatest Bezier'i kõveratest (vt. täpsustavat peatükki Bezier'i kõverast) millel põhineb PS, šrifte konvertides on tavaliselt väikesed vead, ükskõik kumba pidi protsessi teostada. Siiski esineb rohkem vigu konverteerides PS šrifti TT šriftiks kui *vice versa*. Tähtsam on, et vihjamise infot ei ole võimalik konvertida kadudeta kumbagi pidi.

Mõnes artiklis väidetakse, et TT šriftide vektorite salvestamiseks on defineeritud rohkem sõlmpunkte, kui PS šriftidel, või et TT rasteriseerimisprotsess kestab kauem, kuna kasutatav matemaatika on keerukam. Fakt on, et TT matemaatika on lihtsam (ruutvõrrand on lihtsam kui kuupvõrrand), ja mõned tähemärgid on salvestatud vähemate sõlmpunktidega TT šriftis kui PS'is (ringid on talletatud 8 punktiga TT *versus* 12 punkti PS'is). PS rasteriseerimismootor läheb aga tõesti järjest kiiremaks ja on tõsi, et enamus šriftide tehnoloogiaid eelistab pigem väiksemat punktide arvu, kui matemaatilise kirjelduse lihtsust, kuna tänapäeva riistvara võimsuse juures ei ole märgatavat ajakulu vahet arvutuste

tegemisel (olgu siis tegemist ruut- või kuupvõrrandiga). Rõhku pannakse aga endiselt sellele, et šriftide failid oleksid mahult väikesed, kergemini laaditavad üle interneti ja kompaktsemalt integreeritavad dokumentidesse.



Joonis 2. Kolm ringi: liiga vähe (vasakul), liiga palju (keskel) ja täpselt parasjagu (paremal), näitavad kolme moodust defineerimaks ringi. Vasakpoolne kasutab nelja mitte-piirjoonel paiknevat punkti, aga nagu jooniselt näha on seda liiga vähe, et moodustada korrektset ringi. Parempoolne kasutab kaheksat mitte-piirjoonel paiknevat punkti moodustades küllalt perfektse ringi, suhteliselt väikese eksimusega. Keskmine ring on küll korrektne, aga kasutab siiski rohkem punkte, kui oleks mõttekas. Tegelikuses ei anna joonel paiknevad punktid, asudes kurvi tangensite keskpunktis, mingit lisainfot parandamiseks ringi kvaliteeti, võrreldes parempoolse ringiga.

Peamine TT eelis PS1 šriftide ees on fakt, et ta kasutab paremat vihjamise rakendust. PS1 šriftid suudavad vihjata vertikaalseid ja horisontaalseid parameetreid, liita tähemärke, ühtlustada vasturääkivusi ja painutada kõveraid. Mõned neist võivad omada ka erinevaid algväärtusi aktiveeritavate pikslite suuruse kohta.

TT vihjamine suudab teha kõike seda, mis PS rakendus ja kõike muud peale selle, nagu iga paindlik tarkvara. See sisaldab diagonaalide kontrolli, punktide ja tähemärkide piirjoonte liigutamist defineeritud suurustel, jms. Punktide liigutamine teksti erinevate suuruste juures võimaldab šriftiloojatel tööprotsessi käigus käsitsi korrigeerida rasteriseeritud tähemärkide kujutisi.

Järgnev on üks põhimõttelist laadi sümptomitel põhinev erinevus. PS kasutab "rumalaid" šrifte ja "nutikat" tõlgitsejat/interpreteerijat e vihjamist, samas kui TT kasutab suhteliselt "nutikat" šrifti ja "rumalamat" tõlgendajat. Eelnev tähendab, et PS vihjamine ütleb rasteriseerimismootorile, milliseid rakendusi tuleb kontrollida, ja tarkvara teostab seda vastavalt oma intelligentsile – kuidas ja mida. Seega kui Adobe täiendab PS tõlgitsejat, võib vihjamine paraneda.

Vastupidiselt paneb TT väga spetsiifilised instruksioonid (põhimõtteliselt vihjamine, aga mõned TT asjamehed ei armasta seda nii kutsuda) šrifti sisse, kontrollimaks kuidas šrift täpselt genereeritakse. Sellest tulenevalt on šriftiloojatel suurepärane võimalus kontrollida lõpptulemust, kui šrift on rasteriseeritud erinevatel tingimustel. Samas nõuab analüüsimine tõsist jõupingutust, et ka realselt rakendada seda TT vihjamise suurepärase eelist.

Teine faktor on, et TT rasteriseerimine on ehitatud mitmesse operatsioonisüsteemi. Mõlemad, nii Mac OS kui ka kõik praegused Windows'i reinkarnatsioonid, toetavad TT rakendusi otse. Need operatsioonisüsteemid rasteriseerivad TT šrifte ekraanil kuvamiseks ja tegelevad ka konverteerimisega printimiseks, kas siis pikslitena või šriftidena, mida printerid mõnel juhul mõistavad.

Teisest küljest, PS1 šrifti toetus on sisse ehitatud ühele tähtsale operatsioonisüsteemile minevikust - OS/2. See OS toetas PS1 šrifte esimesena nii, nagu teised operatsioonisüsteemid toetavad TT šrifte. ATM'i rasteriseerimismootor peaks olema vastavalt Microsoft'i

ja Apple kokkuleppele ka Windows NT 5.0'le sisse ehitatud. PS šrifid on süsteemipõhiselt toetatud Windows 2000, Windows XP ja Mac OS X operatsioonisüsteemides.

PS šriftide skaleerimine vanemates Mac ja Windows süsteemides nõuab ATM tarkvara, mis haldab rasteriseerimist kuvarile ja rasterdab/konverteerib šrifid mitte-PS printritele. Tehniliselt ei ole vaja ATM'i, et printida PS šrifte PS printeritega, aga ATM on vaja selleks, et kuvada erineva suurusega šrifte arvuti ekraanile.

ATM on kommertstarkvara, teda tarnitakse iga Adobe tootega ja iga Adobe kommertsšrifti koguga. Sellest seisukohast on ta siiani lisand, mitte integreeritud osa operatsioonisüsteemist Mac'i või Windows'i puhul. (Alternatiivina saab kasutada Adobe vabavarana jagatavat Type Manager Light - tarkvara: 4.6 (Macintosh); 4.1 (Windows 95, 98 ja NT 4.0), mis teostab PS rakendusi.

Väiksem, seega parem, võrdlus käib ka šriftide füüsilise salvestamise kohta. TT šriftidel on kogu info ühes failis (tegelikult teeb Windows 3.1 FOT-faili installeeritud TT šrifti jaoks, kui viite originaal-faili asukohale kõvakettal).

Olgu kuidas on, PS šrifid vajavad kahte eraldi faili: üht, mis sisaldab tähemärkide kontuure ja teist, mis sisaldab infot šrifti parameetrite kohta (tähemärkide mõõdustik jms). Macintosh arvutitel vajab PS1 šrift lisaks kontuuršriftile ka pikselkujutistega ekraanišrifti - vähemalt ühes suuruses ja koos infoga šrifti parameetrite kohta. Windows'i PS rakenduste korral sisaldab PFB-fail (*PostScript Font Binary*) kontuure ja PFM-fail (*PostScript Font Metrics*) mõõdustikku. OS'i puhul kannab mõõdustiku infot OFM-fail.

Enamuse šriftidega tuleb ostmisel kaasa ka süsteemist sõltumatu AFM-fail (*Adobe Font Metrics*), selles asuvat šriftimõõdustiku infot võib konverteerida Windows PFM-failiks installeerimisprotsessil ATM'iga, või võib teda kasutada šrifti töötlemise tarkvaraga, genereerimaks ekraanišrifti Mac'i tarvis. (Vahel müüakse Type 1 šrifte ka järgnevas komplektis: PFB-, AFM- ja INF-fail. Sellist šrifti installeerides ATM'iga genereeritakse PFM-fail AFM'ist ja INF'ist.)

Teisest küljest, PS failide paar on sageli väiksem, kui üks TT šrifti fail. Keskmiste šriftide suurus erineb umbes 5%, samas kui TT on umbes kaks korda suurem oma mahuka vihjamissüsteemiga. Samuti kasutavad paljud professionaalsed väljund-seadmed PS keelt oma sisemise graafikakirjelduse baaskeelena. PS šrifte saab nendesse seadmetesse otse laadida, samas kui TT šrifid konverditakse pikselkujutistena, või siis kasutatakse TT rasteriseerimismootorit kui PS programmi, kogu see lisaprotsess võib aeglustada väljatrükki.

Enamusel PS Level 2 printeritel (ja kõigil PS3 printeritel) on sisse ehitatud TT rasteriseerimiskandur ROM'i (ing *read only memory* - püsimälu). Mõne Windows'i printeri tarkvara korral tuli sellele vaatamata muuta printeri seadeid tarkvaraliselt, et seda rakendust kasutada (enamasti tuli endale laadida TT **Type 42**'na, mis on põhimõtteliselt PS programm konverteerimaks TT šrifti infot).

## Edasised praktilised erinevused

Paljud TT tehnoloogia teoreetilised eelised ei ole tegelikult realiseerunud enamuses hetkel müügil olevates TT kommertsšriftides. PS'i toetajad viitavad tänaseni mõnedele vigadele, mis teevad PS šrifid suurele hulgale kasutajatest paremaks lahenduseks kui TT. Tegelikult on mõlematel mainitud väljundkeeltele ühiseid puudusi. On neli praktilist näidet, mis küll kalduvad pigem PS'i kui TT kasuks.

1. Paljud saadaval olevad TT kommertsšriftid, mis on müügil nurgatagustes šriftpoodides komplektina "must-miljon-šrifti-ühe-dollar-eest" on viletsa kvaliteediga. Paljud nendest šriftidest olid algselt vabavara või avalikud PS šriftid, mis on konverteeritud TT'ks mõne lihtsa programmiga. Tähemärkide kontuurid ja vihjamine ei muutu konvertides kohe kindlasti paremaks, kui nad on PS'is, pigem nad deformeeruvad ja kaotavad kvaliteedis, nagu igasuguse automaatse konverteerimisprotsessi puhul. Tavaliselt selliste ekstreemselt odavate TT kolleksioonide puhul ei ole isegi algselt konverteerimiseks kasutatud PS šriftid just parima kvaliteediga.

TT fanaatikud viitavad sellele, et osa šrifte oli saadaval juba enne paljude paranduste ja võimaluste tekkimist - see on lihtne universaalne šriftide skaleerimise süsteem, mis teeb allahinnatud šriftid massidele kättesaadavaks ja loomulikult need šriftid lasti välja kõige laialdasemalt levinud formaadis.

2. Kauga aega ei olnud turul ühtki lõpptarbijale mõeldud šriftitöötlusprogrammi, mis oleks omanud algupärast TT toetust. See küll muutus 1997-dal aastal, kui müügile tuli FontLab 3.0. Kõik eelnenud šriftitöötlus tarkvara kasutas ainult PS põhise Bezier'i kõverate süsteemi, kui oma põhilist tööpõhimõtet. TT šriftide eksportimine sellistest programmidest eeldab konverteerimist ja sellega kaasnevat mõnigaid kontrollimatuid muutusi ja kadusid. Professionaalsed tööriistad TT šriftide töötlemiseks, mis võimaldavad rakendada kõiki tehnoloogia võimalusi ja vihjamist peensusteni, maksavad paraku aga tuhandeid dollareid.

Vaatamata olemasolevatele spetsiifilistele töövahenditele, nõuab esmaklassilise TT vihjamise saavutamise ka täht-tähe haaval käsitsi programmeerimist. See protsess kulub palju aega ja on väga nõudlik töötajate suhtes, kes rakenduse kirjutamist teostavad.

Eelnenu tulemusena on tipp-kvaliteediga TT šriftid saadaval ainult väga suurte tootjate puhul ja isegi siis kasutab vaid murdosa kõigist nendest šriftidest tehnoloogia võimalusi täies ulatuses.

3. TT vihjamise eelised on märgatavad juhul, kui vihjamine on oluline: kui tekste kuvatakse madala ekraanilaotustihedusega seadmetel või muudel madalaresolutsioonilistel seadmetel. Kõrge resolutsiooniga (600 dpi ja enam) laserprinterite laialdane kasutuselevõtt teeb vihjamise ebaoluliseks probleemiks trükitöö puhul. Teisalt jälle laieneb multimeedia ja ekraanigraafika toodang, kus on oluline hea vihjamine, ka internetitoodete puhul tähtsustatakse üha enam korrektset vihjamise protsessi.

4. PS'il on mõned eelised juba lihtsalt seepärast, et ta on olnud kauem turul olev standard, kui TT, eriti käib see tõsisema disaini ja graafika valdkonna kohta. Reklaami- ja kujundusfirmad on standardiseerunud PS'ile ja on teinud selle juurutamiseks suuri investeeringuid PS1 šriftidesse. Enamik šrifte, millel on ekspertrakendused: vanaaegsed numbrid, lisa liittähed, väikesed suurtähed jms on enamasti PS1 formaadis.

Mitmed suuremad šriftidega kauplejad toodavad ka TT šrifte, kuid vaid vähesed (nagu Bitstream) pakuvad oma toodangut paralleelselt mõlemas formaadis. Enamus nagu Linotype ja Monotype konverdivad üksnes mõned oma šriftid TT formeeringusse. Praeguse tarkvara seisuga võtaks isegi tootjate kooskõlastatud tegevuse juures kõigi šriftiperekondade konverteerimine TT formaati mitu head aastat.

Lõpuks palju kuulnud lugu sellest, et mõnigal PS riistvaral, põhiliselt väljatrüki seadmetel, on probleeme TT šriftidega, või kui mitte otseselt TT'ga, siis probleeme on paralleelselt kasutatud erinevate tehnoloogiate puhul, näiteks kui samal lehel ühel tekstireal

on kasutatud nii PS kui TT šrifti. See on enamasti ajalooline põhjus, tingitud tarkvara järk-järgulisest arengust. See oli alles hiljuti, kui TT poogiti külge operatsioonisüsteemidele ja ka uus PS rasteriseerimistarkvara korrigeerib ja parandab pidevalt eelnenud versioonide vigu. Tsiteerides Dov Isaac'i, Adobe kvaliteedijuhti trüki- ja süsteemiosakonnast: "vaatamata sellele, kas sa töötad Mac või PC tüüpi arvutil, võid sa kasutada paralleelselt TT ja Type 1 šrifte hoolimata vastuväidetest, aga sinu süsteemis ei tohi olla mitut täpselt sama nimega šrifti." Ehk kui arvuti-süsteemi on seadistatud kaks identse nimega šrifti (see ei loe, kui üks on TT ja teine PS), võib see vaatamata kõigile tarkvaratootjate püüdlustele tekitada häireid korrektse tööprotsessis.

Töötades Windows keskkonnas, võivad mõned kasutajad avastada, et Windows'i TT **süsteemishriftidest** (ing *system font* - šrifti kasutab operatsioonisüsteem, tavaliselt installeeritakse automaatselt koos operatsioonisüsteemiga) osad asendatakse väljundseadmete poolt PS versioonidega: Times New Roman'ist saab Times-Roman ja Arial'ist saab Helvetica. Kuigi asendus-šriftide põhiline paigutus on sama, ei ole absoluutselt sama konkreetsete tähemärkide vahe (näiteks "v" ja "a" vahe, vms). Kui tarkvara toetab tähevahede korrektuuri (enamasti graafika- ja küljendusprogrammid, ka mõned üksikud paremad tekstiredaktorid), võib tekkida tekstiridade lühenemist/pikenemist, mis tuleb erinevate-aga-peaaegu-samade šriftide kasutamisest arvuti poolt, teostades tekstiküljendust.

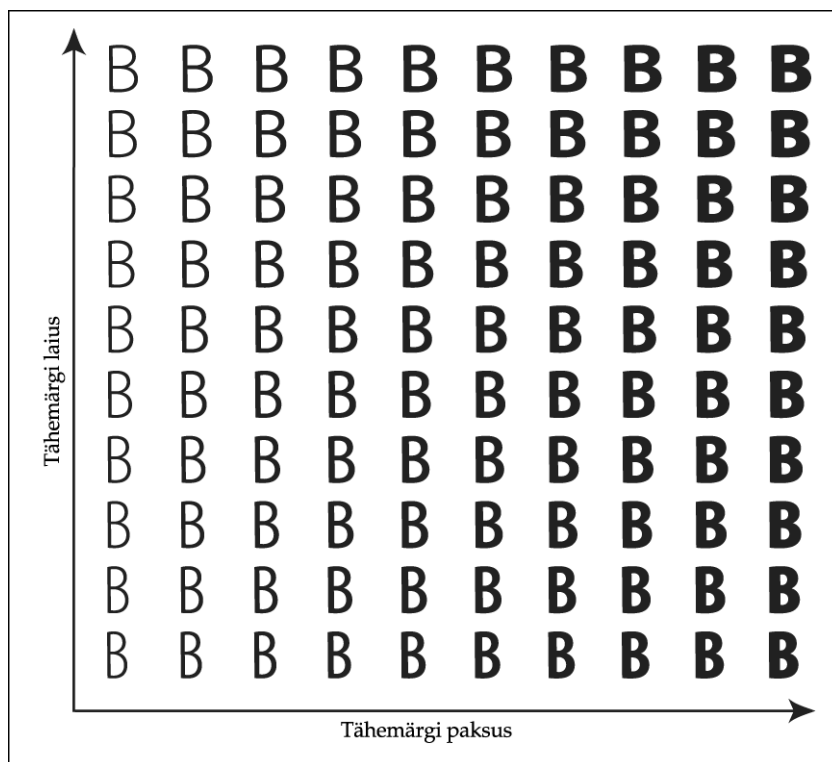
Selleks, et vältida eelnenud näidet, tuleks printeri kontroll-paneelist muuta tingimusi nii, et võime olla kindlad, et kasutatakse süsteemi TT šrifte. Kogenumad kasutajad võivad proovida muuta ka WIN.INI-faili arvutis, see on fail, mis teostab kogu printimise protsessi Windows'is. Kustutada tuleb sealt read šriftide asenduse osas, tagamaks selle, et ekraanil kuvatud TT šrifti kasutatakse ka väljundseadmetes, eelistades seda printeri-poolsele šrifti-asendusele. Windows NT või 95 korral kontrollivad seda protsessi *registri*-seaded. Alternatiiv on ka see, et kasutada oma töös TT šrifti asemel kohe neid šrifte, mida tahaks kasutada printer.

Üks aktuaalne, kuid siiski üsna harva esinev probleem seoses TT'ga, mitte küll sünnipärane šriftile endale, aga seonduv šrifti rasteriseerimisega, nimelt nõuab TT šrifti rasteriseerimine rohkem muutmälu (RAM), kui PS - eriti veel siis, kui TT`de rasteriseerimiseks tuleb laadida spetsiaalne eriprogramm. Kui kasutatavas riistvaras ei ole piisavalt vaba operatiivmälu, võib tekkida probleeme. Šriftide kasutamise kohta on eriti nõudlikud mõned suuremad kujundus- ja disainibürood (nende mure võib sageli olla õigustatud, kuna kokkujooksnud riistvarast tekkivad tihti seisakud töös ja mõningate seadmete puhul ka materiaalne kulu), parim soovitus on konsulteerida enne töö tegemist ja kui nad hoiatavad tekkida võivate probleemide eest, tuleks enne teha mõni test-töö või katsetus, et vältida hilisemaid sekeldusi.

## Multiple Masters

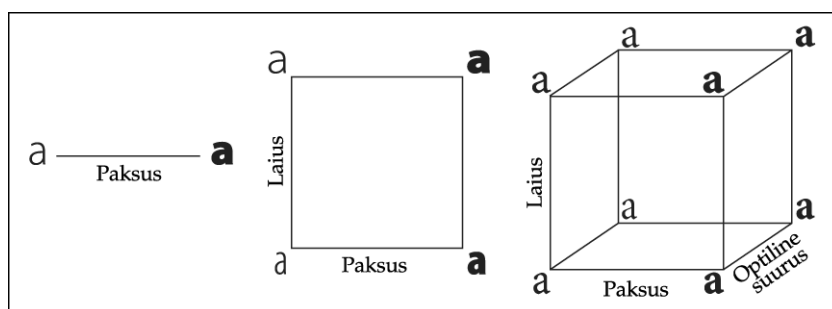
Multiple Master (MM) formaat on täiendus PostScript Type 1 šrifti formaadile. Märkimisväärne on uuendus, kus šrift kasutab tähemärgi loomiseks kaht või enam erinevat tähemärki, millega on tähistatud kujundite jada otsmisi punkte ja vastavalt kasutaja vajadusele genereeritakse nende kahe märgi vahelisest jadast vajalik tähemärk. Seega üks MM-šrift võib omada "paksuse" telge, millel on ultra-peenike tähemärk ühes otsas ja teises ekstra-jäme tähemärk, meetod võimaldab valida kasutajal su-

valise vahepealse tähekuju. Ja see on ainult üks võimalustest, kõiksugu erinevad kujundused võib paigutada MM süsteemi seni, kuni nende Bezier`i kontrollpunktid on võimalik omavahel kokku sobitada.



Joonis 3. Multiple Master šrifti kahemõõtmeline maatriks, mille parameetrid võimaldavad kasutajal defineerida vastavalt vajadusele tähemärgi laiust ja jämedust, joonisel toodud vahepealsete astmete hulk on vaid näitlik.

Samuti on võimalik kasutada mitmeid jadasid paralleelselt, aga iga jada kahekordistab kasutatavate lähtešriftide arvu, kuna iga võimalik otsmine kujund tuleb disainida iseseisvalt. Kujutage ette ruumi nurkadega, kus igas nurgas on lähtešrift. Kui on kolme jadaga MM šrift (ehk kuup) on meil vaja kaheksat lähtešrifti, nelja jadaga šrifti puhul (teoreetiline maksimum) on tegemist aga kuueteist lähtešriftiga, see on ka üks põhjusi, miks sellist šrifti veel genereeritud pole.



Joonis 4. Geomeetriselised kujundid kujutamaks ühe, kahe ja kolme jadaga MM šrifti genereerimise skeemi.

Esmased MM tehnoloogia kasutusrakendused peaksid reguleerima tähemärgi paksust (peenike - paksuks), tähemärgi laiust (kitsad - laiaks) ja optilist suurust (teksti lõppsuurus). Mõned üksikud šriftid eksperimenteerivad ka teiste para-

meetritega, nagu seda on seriifide olemasolu ja suurus jne. Kõiki neid muutusi oleks võimalik teostada ka genereerides lihtsalt uus šrift, aga asja võlu peitubki just siin, MM šrift lubab suhteliselt lihtsalt ja kõrge tüpograafilise tasemega šrifte genereerida kasutajal endal, vastavalt oma hetkelistele vajadustele.

Töö kirjutamise ajal leidsin 36 MM šrifti, mis on avalikustatud suuremate tootjate poolt ja rohkem kui pooled neist on Adobe omad. MM šriftide kasutamine eeldab ATM'i olemasolu, aga seda on PS'iga tegelemisel igal juhul vaja. Mõnedel juhtudel, kui on tege- mist PS Level 1 seadmetega, võib tekkida probleem väljatrüki, mainitud on vanemaid HP ja Apple printereid ning mõningaid PS kloone.

Levinud probleem MM tehnoloogiaga on kasutamise ebamugavus, paljud süsteemid nõuavad iga šrifti ülekäimist ATM tarkvaraga, et võimaldada teda süsteemil kasutada. On küll mõned erandid nagu Microsoft Word ja Adobe PageMaker, aga QuarkXPressi jaoks on vaja juba lisa-tarkvara. Seepärast, ja kuna enamusele kasutajatele on MM tehnoloogia harjumatu, on šriftitootjatel tihti ökonoomsem ja mõttekam lasta kollektsioonid välja mitmetest Type 1 šriftidest koosnevate perekondadena isegi siis, kui nad olid alg- selt disainitud kasutamiseks MM süsteemis. Näide sellest trendist on Jonathan Hoefler'i töötlus šriftist Didot.

Kui oleks olnud parem süsteemi-põhine või tarkvaraline toetus MM-šriftidele, oleksid nad ehk olnud populaarsemad. CompuServe DTP Forum Sysop ja tüpograafia kirjanik Kathleen Tinkel ütlevad: "Kui MM-šrifte oleks olnud kergem kasutada, ma arvan me oleksime näinud teist kahte tosinat šrifti üsna varsti."

## QuickDraw GX & GX šriftid

Järjekordne katse, mis vaimustab tüpograafilise graatsiaga (ja siis...) on Apple QuickDraw GX ja GX šriftid. GX ei ole sarnane ei TT ega PS1 tehnoloogia- le: GX šriftid on edasiminekuks, kus on ühendatud mõlema eelmainitu karakteristika, samuti on seal mõned täiendavad uuendused TrueType GX šriftis.

TT GX šriftidel on samad olulisemad võimalused, kui MM'il tänu analoogsetele vasta- misi disaini telgedele. Telgede kasutamisel on isegi veel suurem paindlikkus. Lähtešrift on võimalik paigutada mõttelise disainiruumi igasse nurka, iga telg nõuab miinimumi- na üht uut tähemärgi kontuuri telje teises otsas, see tähendab, et lähtešriftide kahekor- distamise asemel (nagu MM-tehnoloogias) lisatakse üksikuid lähtešrifte. See teeb või- malikuks kaheksa teljega GX-šrifti, millel on üheksa lähtešrifti, vastukaaluks oleks MM tehnoloogia puhul 256 erinevat disaini. Muidugi disainerid, kes seda tehnoloogiat kasu- tavad, peavad olema ettevaatlikud, kuna disaini-ruumi nurgad peavad olema loodud vektoreid summeerides, mis kipub tihti tekitama probleeme ruumi äärealadel.

Suuremat huvi kogenud kasutajate hulgas on pälvinud GX joondamise ja küljendami- se rakendused (GX Line Layout Manager). Süsteem võimaldab šriftidesse talletatud in- fot kasutades teha mitmesuguseid toredaid asju, näiteks automaatne intelligentne tähe- märkide asendus või optilisel tähekujul, mitte tähemärgi baasjoonel põhinev joondus.

GX-šriftid lubavad kasutada laiendatud kirjamärkide kogumikke, mis on suuremad kui lubatud 256 märki TT ja PS1 šriftide puhul. Need võivad olla alternatiivsed tähemär- gid või mõned märgid, mis tavaliselt asuvad ekspertkogudes, või võõrapärasemates keel- tes või mida iganes kujundaja soovib sinna panna. Vahe laiendatud kogumike ja ekspert- või alternatiivkogude vahel on, et kasutaja saab valida stiililiste erinevuste vahel, mis on

kodeeritud ühte šrifti, seal võib olla keelte eripäradest tulenevaid parandusi, liittähti (ligatuur - kahe tähe ühend) vms. Edaspidi võivad regulaarsed või eritähed ennast ise tänu oma intelligentsile asendada, vastavalt täheühenditele või sellele, kus nad sõnas asuvad.

Kahjuks ei ole GX-šriftid eriti populaarsed. Üheks põhjuseks võib lugeda seda, et tehnoloogiat on võimalik kasutada ainult Macintosh arvutitel, enamus suuremaid rakendustarkvara tootjaid eelistavad aga lahendusi, mis on laiemalt toetatud, ei taheta toetada "standardeid", mis ei ühildu Windows`iga (ja teiste süsteemidega).

GX on mudel, mis tahab ära teha palju funktsioone tekstitöötlus- ja küljendustarkvara eest, see aga ei ole meeltemööda tipp-tasemel kommertstarkvara tootjatele, kes on teinud nendest rakendustest tihti oma tarkvara eelised ja hinnatud väärtused kasutaja silmis. Tarkvaratootjad punnivad kõigest jõust vastu sellele, et näiteks joondamine ja poolitused tarnitakse justkui kandikul kõigile madalama tasemelistele tekstitöötlusprogrammidele, mis lihtsalt nõustuvad toetama GX-tehnoloogiat.

Tulemus on, et üheltki suuremalt tarkvaratootjalt ei ole programmi, mis toetaks täielikult GX-i. On umbes tosin programmi, mis toetab teda osaliselt, nende hulgas kaks küljendusprogrammi Unicorn ja Ready-Set-G. Apple igatahes on lubanud oma süsteemides jätkata GX-tehnoloogia arendamist ja toetamist.

## Unicode

Unicode on rahvusvaheline standard, tähistamiseks avatud kirjamärkide kogu, mis kasutab iga märgi kodeerimiseks kaht baiti. See võimaldab 64000 kirjamärgi kodeerimist 256 asemel ja on vajalik tähistamiseks kõiki tähti maailma kõigis keeltes, igaühel unikaalne ID. Unicode katab siiski ainult keelelised erinevused nagu rühm märkidega tähed jne. Ta ei suuda aga paraku tegeleda tüpograafiliste korrektuuridega, nagu seda on liittähed, vanaaegsed numbrid või väikesed suurtähed. Šrifti disaineri Chuck Bigelow sõnadega: "See võib tunduda kui targutav seisukoht, aga Unicode on pigem kirjamärkide kui tähemärkide kodeering."

Tulemus on see, et Unicode võimaluste lisamine on väga kasulik mitte-inglise või mitme-keelega süsteemide tüpograafia jaoks. Ta ei suuda aga hakkama saada tüpograafiliste rakendustega, mis on suunatud näiteks GX või OpenType rakendustele.

Unicode tähemärkide süsteem on süsteemipõhiselt toetatud Windows NT`s. Hetkel ei toeta Unicode`i ükski teine suurem operatsioonisüsteem. OpenType (mida vaatlen edaspidi) baseerub otseselt Unicode`il ja operatsioonisüsteemid, mis toetavad OpenType saavad Unicode toetuse OpenType funktsiooni siseselt.

## Rahvusvaheline keelte toetus & WGL4

Windows 95 ei toeta Unicode`i, aga tal on vähem universaalne lahendus – rahvusvaheline keelte toetus (*National Language Support*). See lubab kasutada šrifte, millel on rohkem kui 256 tavalist tähemärki, laiendatud Windows`i kogumikes. Erinevate keelte kasutamiseks ja selleks, et vältida ebakorrektsusi vanemas tarkvaras,

peab kasutaja käsitsi valima keele, valitud keele seaded defineerivad aga, millised tähemärgid on kasutatavad klaviatuurilt (eeldusel, et valitud keele kohta on ikka šriftis tähemärgid olemas).

Windows WGL4 kirjamärkide kogu on spetsiifiline rahvusvaheline keelte kogu 652 märgiga, sisaldades kõigi Euroopa keelte tähemärke. See tähendab: kõiki tavalisi ladina tähemärke ja rõhumärkidega tähemärke, pluss kreeka, kirillitsa ja türgi tähestik ja IBM Linedraw. Windows'i baasšriftid (Arial, Courier, Times New Roman) on kõik täiendatud WGL4 tähemärkide süsteemideks. Vaid vähestel teistel šriftidel on sarnane tähemärkide kogum, näiteks on see olemas Microsoft'i Franklin Gothic perekonnal.

## OpenType

1996 aastal valmis Adobe/Microsoft'i koostöös uus šrifti tehnoloogia OpenType, formaat lubab kasutada nii TT kui PS šriftide kontuure ühisel konverteris. Programmid ja enamus operatsioonisüsteeme funktsioneerivad väljaspool šriftide alamsüsteemi, tänu millele töötavad šriftid ühtlaselt kõigis tarkvaralistes rakendustes ja operatsioonisüsteemides. Ühe tehingu osana litsenseerisid Adobe ja Microsoft vastastikku oma tehnoloogiad - TT ja PS'i.

OpenType toetab kõiki uuemaid saavutusi mõlema tehnoloogia TT ja PS formaadist (natuke sarnane GX'le), nagu MM šriftid (PS kontuuridega), multi-keelsed tähemärkide kogud koos täieliku Unicode toetusega ja lisaks veel laiendatud infokogud, toetamaks tüpograafilisi rakendusi, nagu väikesed suurtähed, liittähed, murrud ja alternatiivsed tähemärgid, kõik koos ühes šriftis. OpenType toetab automaatset tähemärkide asendust, nii et ühe tähe võib asendada liittähega (näiteks f-i liittäht vt joonis 18, või paljud araabia tähed joonis 19), või mitmed tähemärgid võib asendada ühega. Tähemärgi asendus võib olla sisutundlik (sõltuv kontekstist) ja/või aktiveeritud kasutaja poolt.

OpenType tehnoloogia südameks on iseseisev platvorm Unicode, rahvusvaheline mitubaiti-kirjamärgi-kohta kodeering, mis katab virtuaalselt kõik maailma keeled. Tulemuseks teeb Open Type mitmekeelse tüpograafia lihtsamaks, lubades mitmete erinevate keelte tähemärgikogusid panna ühte šrifti. Kõigis Adobe OpenType šriftides on standardkogu ladina tähestiku tähtedest, mis on kasutusel kogu lääne-maailmas ja mitmeid rahvusvahelisi tähemärke, kaasaarvatud euro valuuta (€) jms.

Adobe OpenType "Pro" tehnoloogia lisas šriftidesse Kesk- ja Ida-Euroopa keelte (türgi ja poola) rõhumärkidega tähtede täiskomplekti. Mõned šriftid sisaldavad ka kirillitsa ja kreeka tähemärkide kogusid. Adobe pakub ka valikut jaapani tähemärkidega täiendatud OpenType šrifte. Neid šrifte võib iseloomustada sõnaga "Pro", mis on osa šrifti digitaalset nimekujust ning see tähistus ilmub ka kasutaja šrifti menüües. OpenType šriftid, mis ei sisalda Kesk-Euroopa keelte toetust, on märgistatud sõnaga "Standard" ja neid märgistatakse šrifti menüüs nimelisandiga "Std".

OpenType šriftid võivad sisaldada kuni 65 000 tähemärki, muutes olemasolevad tüpograafilised võimalused tunduvalt laiemaks. Paljud ebastandardised tähemärgid nagu vanaaegsed numbrid ja kontekstist sõltuvad või stiililised alternatiivid, eraldi tähekujud sõna alguse ja lõpu jaoks, alternatiivsed ja ilustatud algustähed/initsiaalid ning lisaks suur hulk liittähti võivad kõik olla ühes OpenType šriftis. Varasem tüüpiline PS šrift oli piiratud 256 tähemärgi limiidiga, sundides kasutajaid seadistama kaks või rohkem stiililt sarnast šrifti, kasutamaks "ekspert" või asendustähemärke. OpenType lihtsustab mär-

gatavalt kogu tööprotsessi - tööks vajalike šriftide hulk väheneb ja juhul, kui on vaja kasutada erinevaid riistvara platvorme, ei ole probleeme šriftide ühilduvusega.

Kõik Adobe OpenType šriftid on installeeritavad ja kasutatavad paralleelselt nii PS kui TT šriftidega, neid on võimalik trükkida enamusest väljundseadmetest ja integreerida PDF dokumentidesse. Šriftid on süsteemipõhiselt toetatud Windows 2000, Windows XP ja Mac OS X süsteemides, vanemates operatsioonisüsteemides on vajalik ATM Light, Adobe vabatarckvara. Kui šrift on installeeritud, suudab seda koheselt kasutada enamuse tarkvara, "Pro" rakenduste kasutamiseks peab olema ka tarkvara poolt toetus.

Paljud analüütikud tituleerisid OpenType tehnoloogia Microsoft`i ja TT tehnoloogia võiduks konkurentide üle. See võib olla mingil määral tõsi, kui vaadata tekstitöötlust isolatsioonis. Laiemas perspektiivis võib Microsoft saavutada küll mõjukama positsiooni lõpptaseme kirjastamis/disaini turul, aga see ei ole veel kaugelki tipp. Ei tohi unustada, et uuemates operatsioonisüsteemides on Adobe PS rakendustel süsteemipõhine toetus, tehes Adobe lahendused kättesaadavamaks ja mugavamaks paljudele PS tehnoloogia potentsiaalsetele klientidele. Parim kõigest tundub siiski olema see, et on olemas vastastikkuse toetusega šriftisüsteem, mis vähendab probleeme dokumendi tõstmisel Macintosh arvutist Windows`i masinasse.

(Ironiline, OpenType katse panna TrueType ja PostScript ühisesse formeeringusse on küllalt sarnane sellele, mida Apple tegi QuickDraw GX ja GX-rakendusega šriftidega.)

## Mida toob tulevik?

On huvitav võrrelda OpenType arengut GX`i omaga. GX`i uuendusi ei hakanud aktiivselt toetama ükski juhtiv graafikaprogramm ja tihti on GX`il isegi probleeme muu tarkvaraga ühildumisel. Kas OpenType šrifte saadab suurem edu kui analoogse funktsiooniga GX`i?

GX tehnoloogia entusiastid võivad väita, et kõiki omadusi, mis tänaseks on OpenType tehnoloogial, on võimalik lisada ka GX`ile ja et GX rakenduste puhul on professionaalse tüpograafia toetus lihtsam.

Aga on mitmeid võtme-erinevusi, mis võivad mõjutada OpenType`i tähtsust positiivselt. Esiteks, OpenType tehnoloogia on hästi ühilduv olemasoleva tarkvaraga ja šriftidega - ei ole põhjust, miks peaks muutma kogu süsteemi korraga nagu GX üritas. Teiseks, OpenType on loodud ühe juhtivaima kontoritarkvara tootja (Microsoft) ja juhtiva graafikatarkvara tootja (Adobe) koostöös. Isegi kui ainult nende kahe hiiglasliku kompanii tarkvara toetaks OpenType šrifte, oleks see alustuseks juba palju enam, kui oli GX`i tugi laiatarbe tarkvara hulgas.

Teisalt, tarkvara mis toetab OpenType tehnoloogiat, ei pruugi seda teha nii järgitult, kui GX-põhine tehnoloogia. GX-põhine tähendab seda, et programm laseb kogu töö teha GX-rakendustarkvaral. OpenType puhul peab aga süsteem pöörduma info saamiseks šrifti poole ja info edasine rakendamine on juba iga konkreetse programmi probleem.

Üks asi, mis nõuab mõnede lahenduste puhul heakskiitu, kas siis GX või OpenType, on rahvusvaheliste turgude vajadus. Aasia keeled nagu hiina ja korea, mida ei saa salvestada ja esitada ühebitisel tehnoloogial põhineva šriftina, nõuavad lahendust, mis toetaks suuremat kirjamärkide kogumit. See on väljakutse operatsioonisüsteemide loojatele nagu Apple ja Microsoft ja väga oluline Adobele, kes samuti sooviks hammustada suurt ampsu Aasia turust.

## Kokkuvõte

OpenType võib olla lunastaja šriftide sõjas, aga esmalt tuleks teda toetada operatsioonisüsteemi tasandil ja tarkvara tuleks ümber programmeerida, kasutamaks tema uuenduslikke võimalusi. Samuti võtab aega olemasolevate šriftikollektsioonide konverteerimine uude formaati, parameetrite ühtlustamine/korrigeerimine. Vaevarikas ja töömahukas on ka olemasolevatele šriftidele OpenType tüpograafiliste rakenduste liitmine. See võtab aastaid. Vahepeal on kasutajatel võimalus endal valida.

On mõned erinevused PS ja TT vahel – samuti suhtelisi eeliseid nii ühel kui teisel, mis on aga tihti ühe formaadi aktiivsete fännide poolt üles kiidetud. Mõlemad formaadid – ja kombinatsioon nendest - OpenType – omavad uuendusi ja omadusi, mida ei ole veel laialdaselt kasutusele võetud. Praktikas kasutab enamik mõlemaid formaate ja miksib neid, hoolimata jutust suurte erinevuste kohta ning lõpuks, isegi kui disainer suudab eristada šriftide detaile, ei tähenda see veel, et oleks mingi erinevus lõpptarbija silmis.

Sellest hoolimata on olukordi, kus üks formaat oleks teisest parem, näiteks kui on vaja iseäralikke ekspertparameetreid (enamasti olemas PS šriftides), TT näiteks ei tööta mõnede vanemate tarkvaralahenduste puhul, PS jääb aga viletsaks siis, kui on vaja maksimaalset väljundit kuvaril näitamiseks (parimad TT šriftid).

### *Materjalid*

---

*TrueType & PostScript Type 1* - Thomas W. Phinney - [www.truetype.demon.co.uk](http://www.truetype.demon.co.uk)  
*TrueType Reference Manual* - Apple Computer, Inc - [developer.apple.com/fonts](http://developer.apple.com/fonts)  
*The raster tragedy at low resolution* - Microsoft Corporation - [www.microsoft.com](http://www.microsoft.com)  
*Font formats* - Adobe Systems Incorporated - [www.adobe.com/type](http://www.adobe.com/type)  
*What's the difference between all these font formats?* - Norman Walsh - [www.nwalsh.com/comp.fonts](http://www.nwalsh.com/comp.fonts)

## **True Type - šriftide ajalugu**

1980-datel taipasid enamus toleaegseid PC (personaalarvutite) turul tegutsejaid, et skaleeritavate šriftide tehnoloogia kujuneb vajalikuks komponendiks tuleviku operatsioonisüsteemide osana. Adobe üritas sundida Apple't ja Microsoft'i selleks otstarbeks litsenseerima oma PS koodi. See lahendus (millest sai Display PostScript - DPS) omas vajalikke tehnilisi omadusi, aga mõlemad kompaniid olid loomulikult liiga kitsid loovutamaks võtmeosade kontrolli oma operatsioonisüsteemidest - rääkimata miljonitest "kuninglikest dollaritest" - Adobe kasuks. (DPS oli selleaegsetele masinatele liiga aeglane ja kohmakas, töötades korralikult ainult Steve Jobs'i NeXT masinatel.) Samuti oli Apple häiritud sellest, et Adobe litsenseeris oma PS tehnoloogia otse printerite tootjatele, vähendades sellega Apple LaserWriter tehnoloogia kasutamist. Niisiis sõlmisid Apple ja Microsoft ristlitsenseerimise ja tootearenduse lepingu, tehing oli soodus mõlemale poolele: Microsoft käib välja PS stiilis graafika mootori (TrueImage), samas kui Apple loob šriftide süsteemi, mis oleks isegi parem kui Adobe oma...

TrueImage oli täielik läbikukkumine. See oli vigane, kui seda hakati tarnima ja Microsoft ning Apple mõistsid, et neil ei olegi seda tegelikult väga vaja. (Selgesti on näha, kes lõi kokkuleppes rohkem kasu!) Programmide loojad tahtsid teha asju omamoodi, mitte jääda toppama kirjutades kloone Adobe Illustrator'ile. (Aastaid hiljem laskis Apple, kelle kliendid olid nõudlikeimad töödeldava graafika kvaliteedi suhtes, välja interaktiivsele graafikale ja tüpograafiale palju paremini sobivama süsteemi QuickDraw GX.)

Adobe võttis kaotust väga tõsiselt. See nurjas auahne plaani kontrollida kogu PC-šriftide tehnoloogiat. Selle mõju oli kahekordne. 1989 aasta keskel, kui Adobe mõistis, et Apple süsteemid on avatud teiste tarkvaratootjate lahendustele, teavitas ta oma uuest programmist Adobe Type Manager (ATM), kusjuures avalduse tegemise hetkel ei olnud programmi veel kirjutama hakatudki. Umbes aasta hiljem oli tarkvara turul ja iga soovija võis selle osta, et kasutada Adobe Type 1 šrifte Macintosh masinates, ilma igasuguse Apple abita. ATM'i müüdi odavalt, või siis jagati tasuta koos Adobe'st ostetud šriftide ja muu tarkvaraga. Teine suur samm oli Type 1 šriftide tehnilise spetsifikatsiooni avaldamine. (Eelnevalt pidid šriftide loojad maksma autoritasu Adobe'le, et luua oma Type 1 šrift. Šrifti info krüptiti ja ei olnud võimalik tuvastada selle strateegilisi kontrollpunkte, välja arvatud sügaval sügaval Adobe PS graafikamootori sisemuses. Aga nüüd võis iga soovija kirjutada Type 1 šriftide genereerimise või muutmise rakenduse/tarkvara.) Fakt on, et tegelikult oli Adobe selle sammu tegemiseks surutud vastu seinale juba hetkel, kui TT šriftide karakteristika tehti avalikuks ja kuna ka Bitstream oli niikuinii muukinud lahti Type 1 formaadi. Bitstream laskiski varsti turule sajad Type 1 šriftid ja kiire ATM'i kloni - FaceLift.

Apple alustas oma TrueType tehnoloogia arendamist 1987-dal. Selleks ajaks oli turul ohtrasti erinevaid, konkureerivaid skaleeritavate šriftide tehnoloogiaid ja mõned nendest olid kohandatud ka Macintosh arvutitele. See ei olnud mingil juhul kindel, Sampo Kaasila Macintosh'i inseneri sõnul, et Apple adopteerib TT tehnoloogia. Lõpptulemuseks tõestas tehnoloogia ennast ise, olles optimaalne töötamiseks ja parim renderdamisel (nii suurte, kui ka väikeste resolutsioonide puhul) võrreldes teistega. Kaasila lõpetas oma

töö TT'ga augustis 1989, siis ei kandnud see tehnoloogia aga veel seda nime. Järgneval kuul kuulutasid Apple ja Microsoft välja ühise strateegilise koostööprojekti vastukaluks Adobe'le. Projekti käigus pidi Apple looma šriftisüsteemi ja Microsoft printimise tarkvaralise lahenduse. Apple esitles oma TT maailmale mais 1990, kui avalikustati System 7 - tehnoloogia tuum oli üldjoontes sama, mille Kaasila oli kirjutanud 1989-ndal aastal. Tegelikult oli esimene kliendi versioon 80 kilobaiti, lisatud System 6.0'ile, saadaval veel praegugi Apple kodulehelt! Süsteem vajab iseenesestki mõista šrifte ja esimesed TT šriftid - Times Roman, Helvetica ja Courier - olid suurepäraseks näiteks sellest, mida uus tehnoloogia võimaldas. TT šriftid on alates sellest hetkest Mac'ide operatsioonisüsteemi lahutamatu osa.

Microsoft tutvustas TT šrifte 1991 aprillis, kui ta avalikustas Windows 3.1 versiooni. Töötades koos Monotype'iga, olid nad loonud baasšriftide valiku - TT versioonid šriftidest Times New Roman, Arial ja Courier. Need šriftid tõestasid, samuti nagu Apple TT'd, et muudetava suurusega šriftid suudavad genereerida bitmap-tehnoloogial põhinevaid kujutisi virtuaalselt, ja seda küllaltki edukalt.

Lihtsate šriftidega töötasid arvutid üldiselt korralikult. Alates Windows 3.1'st, mis jooksis aeglase 16-bitise 286 protsessoritega, tuli TT süsteem ümber häälestada kui 16-bitine versioon Kaasila originaalselt 32-bitisele arhitektuurile mõeldud versioonist. (Kuigi 32-bitine simulatsioon töötas, arvasid mõned, et see on liiga aeglane sellele arvutite põlvkonnale.) Mälu jagamine töötas ainult "enamuse ajast". Suurte šriftide puhul muutus kogu süsteem ebatäpseks, siiski polnud see kõige suurem probleem. Mõnede šriftide puhul, mis olid karakteristika järgi usaldusväärsed, ei õnnestunud kuvada keerukamaid tähti, või olid nad kuvaril, aga ometi oli neid võimatu printida. See oli müsteerium ja raevutsema ajav probleem kõigi šrifti-kasutajate jaoks üle kogu maakera.

Asjad olid pahad ka šrifti loojatele. Keerukat koodi tuli lihtsustada. Vihjamise kood tuli ümber kirjutada 16-bitisele arhitektuurile. Šriftid, mis töötasid edukalt Mac'il, loodud TT šriftide genereerimise tööriistadega (kõik Mac'i programmid, enamuse on kirjutanud Kaasila), keeldusid töötamast Windows süsteemis. Rahulolematuse põhjuseks oli ka TT vihjamise raskus masinale ja mitmed šriftiloojad loobusid varajasematest lubadustest konverteerida oma šrifti-kataloogid TT formaati. (Isegi nüüd on šriftide loojatel mitmeid šrifte ootamas suurklienti, kes ütleks: "Olgu ma võtan 10 000 litsentsi" see oleks õiglane, tasustamaks mitmeaastast tööd TT tehnoloogia väljatöötamisel.) Niisiis juhtuski, et peamised ja suuremad šriftide genereerimise tööstused jätsid TT'de turu amatööride meelevalda. Turg oli peagi üle ujutatud odavate šriftidega, skaneeritud ja viletsalt konverditud teiste inimeste tööst - tihtilugu vigased, ebakorrektsed piirjoontega ja kasutu vihjamisrakendusega. Märgatavalt vähenes kvaliteet ja sellega kannatas teenimatult kogu TT šriftide maine, kui mitte kõikjal, siis kindlasti Windows'i platvormil on TT kujunenud professionaalide hulgas põlatuks ja ebaotstarbekaks.

Augustis 1995, kui turule jõudis Windows 95, sai Microsoft'i TT 32-bitise graafika mootori, täieliku ja usaldusväärse. Tõepoolest, see suudab genereerida must-valgeid pikselrastreid ja teostada tähemärgi piirjoonte korrektuuri. Tänapäevaks on Microsoft teinud ka mõned suurepäraseid TT šriftide prerakonnad, valik šrifte on tasuta kättesaadava ka ametlikul Microsoft'i veebilehel.

Microsoft'i jätkuva TT arendustöö edusammude hulka kuulub ka TrueType Open šrift. TT Open töötab kõigil Microsoft'i platvormidel ja Apple Macintosh masinates, tehnoloogial on multi-keeleline toetus ja kõrgetasemelised tüpograafilised omadused.

TT Open formaadile järgneb täiustatud versioon TrueType Open 2 ehk OpenType, Adobe Systems'iga koostööst sündinud toode, mille formaat võimaldab kasutada nii PS kui TT (ja TT Open) infot šriftide genereerimiseks.

# Kommentaarid Greg Hitchcock'ilt

*Greg Hitchcock on üks Microsoft`i inseneridest, kes adapteeris Apple koodi Windows`ile, ja kes täiustab Microsofti süsteemi alates sellest ajast.*

Ma ei nõustu väitega 16-bitise rasteriseerimise tehnoloogia kohta Windows 3.1 puhul. Tehniliselt on kõik korrektne. See oli kohutav viga muuta kodeering 32-bitilt 16-bitile ja ma olin sellele algusest peale vastu. Kui me oleksime matkinud 32-bitist süsteemi 16-bitiga, poleks see nii palju ja nii tõsiselt mõjutanud arvuti korrektset tööprotsessi. Enamus probleeme oli 16-bitise konverteerimisprotsessiga ja seda üritati lahendada kahe paralleelse konverteerimismootoriga Windows 3.1 puhul - 16 ja 32-bitisega. 32-bitine pidi hakkama tööle, kui 16-bitine jääb hätta. 16 biti puhul polnudki suuri probleeme šrifti ekraanile kuvamine, enamus kuvamise probleeme suudeti lahendada tiheda tööprotsessi käigus juba enne tehnoloogia ametlikku presentatsiooni. Windows NT 3.1 puhul kasutasime juba ka 32-bitist rasteriseerimiserakendust. Ja rasteriseerimis-rakendus, mis tuli müügile Windows 95`ga oli juba totaalselt ümber kirjutatud, see polnud küll väliselt märgatav (väljaarvatud sujuvad täheservad - *anti-alias*), aga tal olid seesmiselt suured arhitektuurilised muutused.

Me pingutasime kõvasti, et Windows`i baasšriftid oleksid edukad. Šriftide disainimiseks ja täiustamiseks kulus umbes kaks aastat. Kui oli esimene avalik šriftide presentatsioon 1991 aastal NewYork`is Monotype`i konverentsil, valitses kõigi tarkvara esindajate ja teiste koosolijate hulgas üksmeel, et nähtu oli parim ekraanšrift, mis eales tehtud. See on ilmselt ka väitluse põhjuseks, miks ma töötasin projektiga aastaid ja ma nägin tehnoloogia puudusi ja ma pettusin meie sisemises vihjamises rakenduses - mitte isegi rakenduse tööprotsessis, vaid selles, kuidas tulemus saavutati.

Peale tehnoloogia avalikustamist toimus plahvatuslik Windows`i šriftide tootmise kasv. Kõik suuremad kompaniid käisid välja oma kollektsioonid. Esimene Microsoft`i enda toodetud šrifti-kogu töötas uskumatult korrektselt, shokeerides isegi Microsoft`i juhtkonda.

Lõpuks korjas teema üles ka press. Umbes 6 kuud enne Windows 3.1 tarnimise algust, tegi meie töögrupp tõsiseid edusamme, harides pressi TT võimalustest ja sellest, milliseid suurepäraseid šrifte on nüüd võimalik toota. Oli palju demosid, mis võrdlesid meie šrifte Adobe omadega. See kõik aitas kaasa nn "šriftide sõja" puhkemisele. Tegelikult me ei märganudki erilist vastukaja oma edule - enne kui Windows 3.1 lõpuks turule jõudis, siis alles hakkas ilmuma virnade viisi kiitvaid artikleid TT šriftidest ja Windows`ist. Huvitav oli seegi, et press omistas TT tehnoloogia loomise au Microsoft`ile, mitte Apple`le. See ei olnud otseselt meie süü, kui välja arvata, et meie ülistasime TT taevani, samas kui Apple ei öelnud midagi.

Kõik see, mille lõi TT ja Windows 3.1, sai aga tugeva tagailöögi. TT šrifte on väga raske korrektselt genereerida, aga ometi tahtsid selles mängus osaleda nii paljud. Liiga suuri lootusi pandi nn automaatsele šriftide loomise tarkvarale. Minu arvates see viiski suurte kvaliteedi-erinevusteni, mis esinevad tänaseni TT šriftides.

## *Materjalid*

*A History of TrueType - TYPE chimérique - [www.truefont.demon.co.uk](http://www.truefont.demon.co.uk)*

*A brief history of TrueType - Microsoft Corporation - [www.microsoft.com/typography](http://www.microsoft.com/typography)*

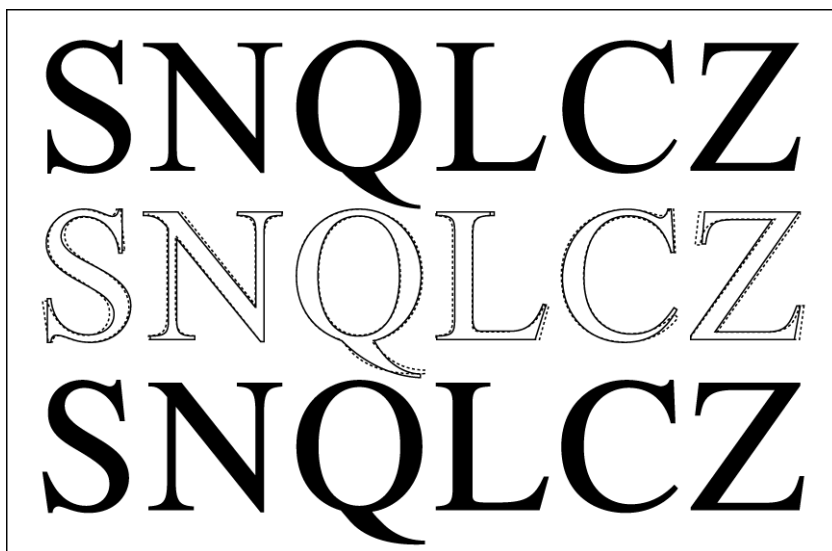
*A talk with Sampo Kaasila - TYPE chimérique - [www.truefont.demon.co.uk](http://www.truefont.demon.co.uk)*

*What is the History of TrueType? - David K. Every - [www.mackido.com](http://www.mackido.com)*

## Times (New) Roman

Tähestiku on loonud "The Times" ajaleht oma isiklikuks tarbeks, Stanley Morison'i juhtimisel. **Times Roman** on nimi, mida kasutas Linotype, ja nimi, mille disainile nad registreerisid kaubamärgi USA's. **Times New Roman** oli ja on siiani nimi, mida kasutab Monotype Kompanii. Tähestik on algselt loodud Monotype poolt Inglismaal, kujundus oli litsenseeritud aga Linotype'le, kuna ajaleht "The Times" kasutas Linotype'i seadmeid oma igapäevases töös.

Teise Maailmasõja ajal andis Ameerika Linotype kompanii, üldises liitlaste seltsimehelikus vaimus, avalduse kaubamärgi registreerimiseks nimele Times Roman, nagu oleks see tema šrift, mitte Monotype'i või The Times'i oma. Registreerimine leidis aset 1945 aastal. 1980 vaadati kaubamärgi registreerimine uuesti üle, kuna mõned eraettevõtjad ihaldasid nime kasutamisest saadavat tulu, mida sai Rupert Murdoch ("The Times'i" omanik). Algatati eraldi juriidiline protsess selgitamaks Monotype'i õigusi, kasutamaks šrifti nime USA's, hoolimata Linotype'i litsentsist. Juriidiliste manöövrite tulemus on, et Linotype ja tema litsentsid Adobe ja Apple'ga Times Roman'i nime kasutamise kohta jäid jõusse, samas jäi Monotype'le Times New Roman ja litsents Microsoftiga.



Joonis 5. Kasutusel olevate Times (New) Roman versioonide pisikesed erinevused. Ülemine kiri on Monotype/Microsoft'i OpenType formaadis Times New Roman, keskel šriftide erinevused, alumine šrift - PS formaadis Linotype kompanii Times.

Aastaid kestnud üleookeanilise Times'i disaini jagamisel ja trükitehnoloogia üleminekul metallplaatidelt fotole ja edasi digitaalseks, toimusid suured muutused šrifti detailides nii Linotype kompanii Times Roman'is kui Monotype kompanii Times New Roman'is. Eriti ilmsiks tulid need erinevused siis, kui Adobe laskis välja oma PS versiooni Times'ist. Erinesid tähemärkide kõrgused, samuti proportsioonid ja mitmed detailid.

Hilistel 80-datel uuendas Monotype oma Times New Roman'it, kohandades selle vastama nii proportsioonidelt kui mõõtkavalt Adobe-Linotype'i versioonile Times Roman'ist. Monotype väitis, et nende versioon on Adobe omast parem, kuna sellel on pehmemad kurvid, täpsemad detailid ja üldse tundlikum algupärasem disain, mille tegi Victor Lardent spetsiaalselt "The Times'ile" ja Monotype'le. Lardent oli töötanud Stanley Morison'i töögrupis. Samal perioodil täiendas Adobe oma versiooni Times'ist, kasutades samuti disainereid Linotype kompaniist. Peale täiendust tuli loomulikult pressiteade, mis ülistas suurepärasest uut Adobe Times'i versiooni. Seega käis turul omamoodi rebimine, kes loob kõige rafineerituma, tundlikuma, originaalsema, tõepärasema, siirama ja tüpograafiliselt korrektsema šrifti. Paljud, isegi enamus, kasutajaid ei märganud ega hoolinud imepisikestest erinevustest, paljud muudatused olid nähtavad 10 punktise teksti juures 300 dpi resolutsiooniga (muudatus oli näiteks 1 piksel pikem seriif vms).

Kui Microsoft tootis oma versiooni Times New Roman'ist, litsentseeritud Monotype'lt TT formaadis ja Apple valmistas oma versiooni Times Roman'ist, litsentseeritud Linotype'lt samuti TT formaadis, algas uus võistlus uuest aspektist, kuna nii Microsoft kui Apple olid kulutanud kõvasti aega ja jõudu tegemaks TT šrift nii heaks, kui võimalik, või paremaks veel, paremaks isegi kui PS šriftid. Samal ajal avalikustas Adobe oma ATM'i, millega kaasnesid ka täiendatud baasšriftid, kohendatud põhiliselt perfektsemaks kuvamiseks ekraanil. Ka Imagen, nüüdseks osa QMS'ist ja Sun'ist, arendas oma konkureerivat skaleeritavate šriftide tehnoloogiat ja ponnistas tõestamaks, et nende versioon Times'ist, litsentseeritud Linotype'lt, ei jää sugugi alla konkurentide omadele. Sellest tulenevalt muutus Times'i kvaliteet märkimisväärselt, Times'ist kujunes etalon paljudele šriftiformaatidele.

Ei kunagi varem ja ilmselt ka mitte edaspidi, ei saa pikslite paigutus seriifides või "Sköver" vms nii suurt tähelepanu inseneride ja arvutiteadlaste hulgas, kui seda oli Times Roman'i puhul. See oli ülim ajastu digitaalsete šriftide loojatele.

#### *Materjalid*

---

*Times (New) Roman and its part in the Development of Scalable Font Technology*  
- By Charles Bigelow - [www.trueType.demon.co.uk/articles](http://www.trueType.demon.co.uk/articles)  
*Times - Apple Computer, Inc* - [developer.apple.com](http://developer.apple.com)  
*Times New Roman - Microsoft Corporation* - [www.microsoft.com/typography/fonts](http://www.microsoft.com/typography/fonts)  
*Times Roman Linotype 1931-1935 - MyFonts.com* - [www.myfonts.com](http://www.myfonts.com)

Neljas peatükk. Vaatlen šrifti tehnoloogiale olulisi mõisteid, esiteks šrifti kontuure (ing. outline) e väliseid piirjooni.

# Šrifti kontuurid

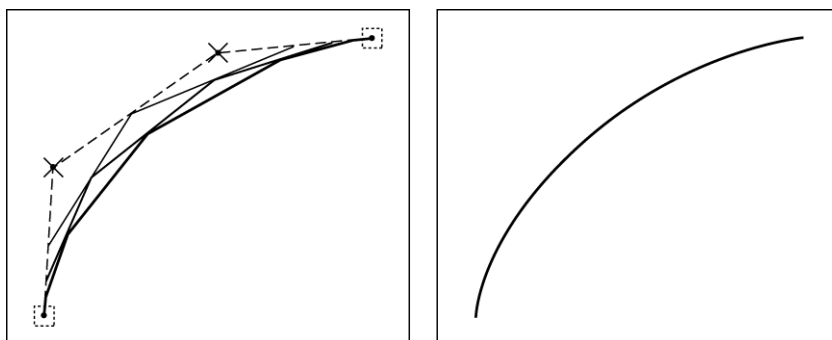
## BEZIER'i kõver

*Bezier'i kõvera autoriks on Prantuse insener Pierre Etienne Bézier (1910-1999). Bézier kasutas kõveraid Renault autokerede konstrueerimisel 1970-datel.*

Suure resolutsiooniga jooniste/grafika ja teksti kujutamisel tagab parima tulemuse vektorgraafika. Vektorgraafika võimaldab esitada kujutisi nii suuremalt kui väiksemalt, säilitades kujutiste sujuvad piirjooned. Vektorite eeliseks on, et neid on lihtne salvestada – infomaht on võrreldes pikselpindadega väike. Kõik vektorid ja kuvamiseks vajalikud punktid on määratletud koordinaatide ja kindlate valemitega.

Bezier'i kõver on väga levinud vektorite arvutamise meetod. Kõvera joonistamiseks kasutatakse kahte tüüpi punkte. Sõlmpunktid, mis asuvad alati vektoril ja kõveral mitte asuvad punktid, mille poole tavaliselt kõver kaardub. Et kujutada pikemat kõverat, on vaja ühendada mitu Bezier'i kõverat. Tavaliselt tehakse siis ühe kõvera lõpppunkt teise alguspunktiks. Kurv läbib iga lõigu algus- ja lõpppunkti, mis teeb kõvera töötlemise lihtsaks ja meetodi enda populaarseks paljude vektorgraafika programmide töös.

Bezier'i kõvera idee on küllaltki lihtne. Et kujutada pehmet ja sujuvat kõverat, ühendatakse esmalt sõlmpunktid vektoril mitte asuvate punktidega kurvi kaudsel trajektooriga. Seejärel joonistatakse sirged, ühendamaks eelmiste sirgete keskpunkte ja seda korratakse seni, kuni joon on piisavalt sujuv. Kõvera joonistamine on kujutatud alltoodud joonisel.



Joonis 6. Bezier'i kõvera konstrueerimine ja kõver meie nähtuna. (Ruudud tähistavad kõvera joonel asuvaid sõlmpunkte, ristid kõveral mitte asuvaid punkte.)

Vektorkujutist on võimalik defineerida ja salvestada, kirjeldades ainult sõlmpunktide ja kõveral mitte asuvate punktide koordinaate.

*Materjalid Bezier'i kõvera kohta*

*2-D Curve Generation - Darren Meyer, WPI CS Department - [www.cs.wpi.edu](http://www.cs.wpi.edu)  
Bezier curves - Luc Devroye, School of Computer Science, McGill University - [cgm.cs.mcgill.ca](http://cgm.cs.mcgill.ca)*

# Vektorite matemaatika

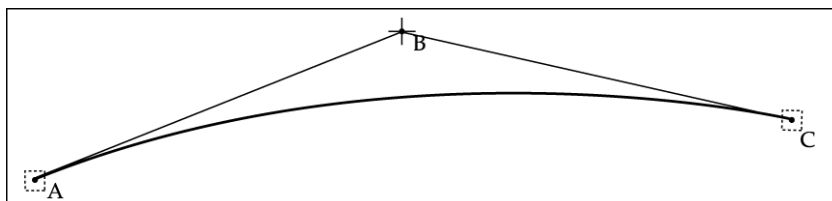
Vektorid koosnevad enamasti B-kõveratest, mis üksteisega ühendades moodustavad Bezier'i kõvera. Iga B-kõver on defineeritud kolme piirjoone kontroll-punktiga, genereeritud vastavalt parameetritele, piirjoonte rasteriseerimiseks kasutatav võrrand on šrifti sisse programeeritud.

Kui kolm punkti kõveral on  $(A_x, A_y)$ ,  $(B_x, B_y)$  ja  $(C_x, C_y)$ , siis:

$$p_x = (1-t)^2 \cdot A_x + 2t(1-t) \cdot B_x + t^2 \cdot C_x$$
$$p_y = (1-t)^2 \cdot A_y + 2t(1-t) \cdot B_y + t^2 \cdot C_y$$

Muutes parameetrit  $t$  0-ist 1-ks genereerime kõik punktid  $p$  kõveral, mis on defineeritud parameetritega  $A$ ,  $B$  ja  $C$ .

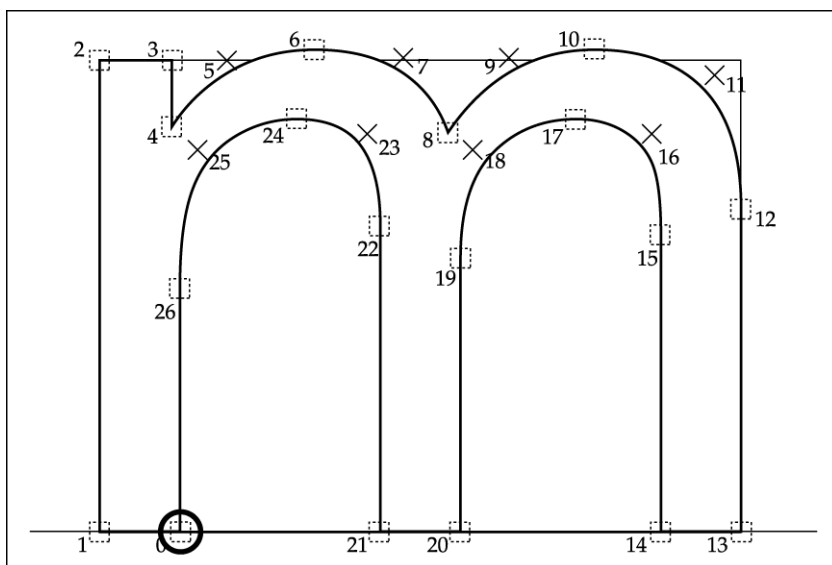
Visuaalselt, kõver jookseb kahe kõveral asuva kontrollpunkti  $A$  ja  $C$  vahel, kasutades joonel mitte asuvat kontrollpunkti  $B$ , et mõjutada joont kõverduma tema poole. Kõver väljub punktist  $A$  piki sirget  $AB$  ja siseneb punkti  $C$  piki sirget  $BC$ . (See teeb mugavaks ja lihtsaks nurga defineerimise ja kasutamise tangensi abil.)



Joonis 7. Konstrueeritud kõver. (Ruudud tähistavad kõverate joonel asuvaid sõlmpunkte, ristid kõveral mitte asuvaid punkte.)

Joonisel olevad punktid  $A$ ,  $B$  ja  $C$  vastavad eeltoodud võrrandile ja defineerivad ühe lõigu tähemärgi piirjoonest. Võimalik on, et mitmed kõveral mitte asuvad punktid on üksteise järjel, sellisel juhul tekib lõikumine tähe piirjoonega täpselt sirge keskel.

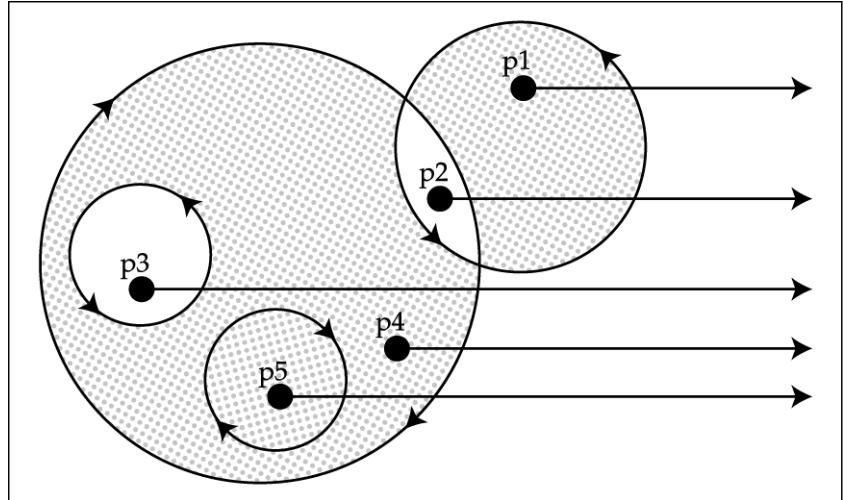
Tähemärgid on defineeritud nagu kontuuride seeriad, iga seeria koosneb kolmest või rohkemast punktist. Iga punkt on kas joonel asuv või väline punkt. On õimalik, et kontuur koosneb ainult kontuurivälistest punktidest.



Joonis 8. Täht "m" šriftist Monotype Arial. Kontuuride seeriad: 4 - 6; 6-8 jne.

# Täispinnad paremale

Hooletult tehtud šriftide probleemiks on sageli see, et kujundid lõikavad iseennast või kattuvad üksteisega, paljud tarkvaralised lahendused ei suuda sellisel juhul neid hallata, või teevad seda vigaselt. Korrektses šriftis on täispinnad defineeritud nii, et sõlmpunktid asuvad piirjoonel päripäeva. Täispindades välja lõigatud kujundid defineeritakse vastupäeva.



Joonis 9. Vastavalt sellele, kas piirjoon on päripäeva või vastupäeva, defineeritakse tema pind täispinnaks või auguks teises kujundis. Samuti saab skeemi vaadates küsida, kas näidatud punktid asuvad kujundi sees või väljas? Otsustamiseks kontrollitakse, kas punktil on pöörlemise suund või ei.

- Punktid p1 ja p4 asuvad ühe kindla kujundi sees, see annab neile mingi kindla suuna - kas siis päripäeva või vastupäeva, tähendab nad on kujundi sisesed punktid.
- Punkt p2 asub nii vastu kui päripäeva kujundi sees, seega liikumised liites saame suunaks null ja punkt on väline.
- Punktid p3 ja p5 illustreerivad seda, kui oluline on, et piirjoontele suund oleks korrektselt defineeritud. Punkt p3 puhul nullib piirjoonte vastastikune suund jällegi punkti suuna - tegemist on välise punktiga. Punktil p5 aga liituvad liikumised ja seega on punktil kindel suund ja ta on kujundi sisene punkt.

On olemas kolme liiki tähemärke:

- **Lihtsad** tähemärgid sisaldavad lihtsalt oma piirjooni.
- **Kombineeritud** tähemärgid viitavad teistele tähemärkidele, et

kombineerida oma piirjooned mitme tähemärgi piirjoontest. Viidatud komponendid võivad olla lihtsad või samuti kombineeritud. Tavapäraseim kombineeritud tähemärk on näiteks rõhumärgiga täht.

- **Null-kontuuriga** tähemärk, nagu näiteks tühik, tähistab mingeid omadusi tekstis, aga ei oma trükitavat väljundit. Null-kontuuriga tähemärkidel, nagu kõigil tähemärkidel, on vasakpoolne joondus, aga nende parameetrid ei ole muudetavad nii nagu nad on muudetavad tavaliste tähemärkida puhul.

*Materjalid*

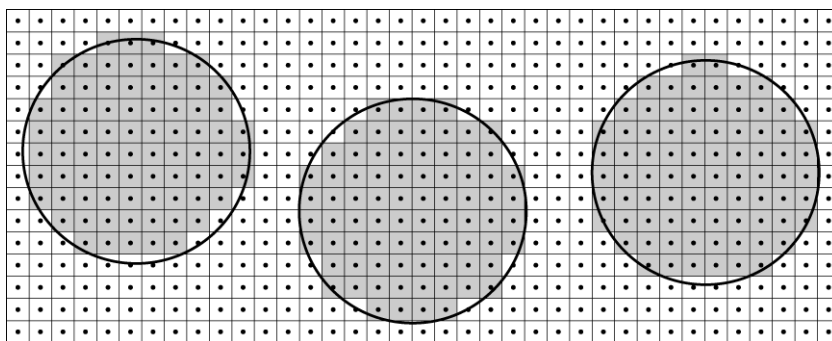
*AAT Font Quality Specification - Apple Computer, Inc. - [developer.apple.com/fonts](http://developer.apple.com/fonts)  
TrueType Outlines - TYPE chimérique - [www.trueType.demon.co.uk](http://www.trueType.demon.co.uk)  
Converting Outlines to the TrueType Format - Apple Computer, Inc. - [developer.apple.com/fonts](http://developer.apple.com/fonts)*

Viies peatükk. Kui tegemist on digitaalsete šriftidega või digitaalse tekstitöötlustarkvara, tuleb kindlasti tegemist ingliskeelse mõistega “hinting”, vahel kasutatakse paralleelselt ka lihtsalt sõna “juhendid” (ing. instructions), tegemist on niisiis juhendiga – käsustikuga šrifti korrigeerimiseks ja rasteriseerimiseks. Kasutan siin ja mujal töös “hinting” kohal sõna vihjamine.

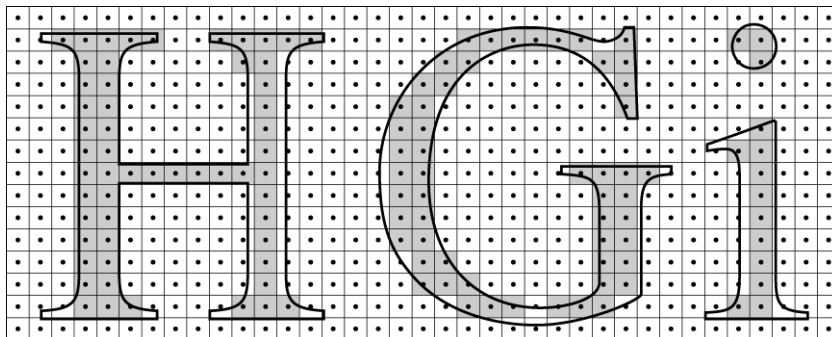
## Vihjamine

Alates hetkest, kui digitaalsete šriftide ja tarkvara arendajad / tüpograafid asusid lahendama skaleeritavate šriftide rasteriseerimise probleemi madala ekraanisagedusega seadmete tarvis, teadvustasid nad endale ka vihjamise komplitseerituse. Skaleeritavate šriftide puhul on see probleem märgatavalt tõsisem, kui ülejäänud muudetava graafika puhul, kuna tekst nõuab süvenemist ja inimese silmad väsivad kiiresti, kui ta peab lugema ebaühtlast teksti.

Matemaatiliselt korrektne pikslite esitus tähemärgi erinevate suuruste ja madala ekraani lahutusega kuvarite juures ei ole tihti visuaalselt ilus, või on suisa mitteloetav. Vihjamine on käsustik, koondamaks tehnikaid, mis peavad tagama teksti loetavuse ja maitsekuse nii palju, kui see on parajasti võimalik.



Joonis 10. Näide sellest, kuidas bitmapkujutis muutub vastavalt täidetava piirjoone paigutusele ekraaniresolutsiooni suhtes. Ideaalne rasteriseerimine tähendaks aga seda, et täidetud piirjoon on alati (loe: ühe ekraanilaotus sageduse ja suuruse juures) sama kujuga, omadused nagu sümmeetria ja proportsioonid peavad aga kindlasti säilima ka suuruse muutudes. Selline ebakorrektno rasteriseerimine nagu joonisel põhjustabki ebatäpseid tähemärke, seriifide kadumist ja erineva paksusega šriftide teket.

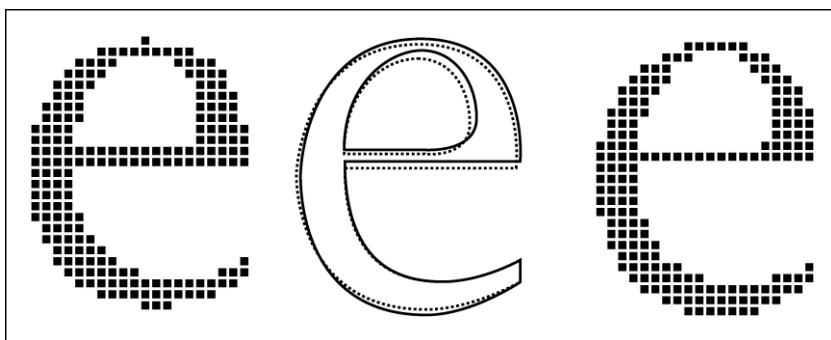


Joonis 11. Näha on rasteriseerimisest tulenev tähemärkide ebakorrektsus, “G” alumise kaare puhul on tegemist “ära jäämise” (ing. drop out) efektiga e tähemärk muutub katkendlikuks.

Vihjamine kätkeb endas hoolikalt tehtud ja sageli imepikesi korrekture tähemärgi piirjoone täitmise protsessi muutmiseks. Probleemid madalal resolutsioonil - tähed kaovad osaliselt, mõned detailid ilmuvad paksendatult või peenemalt, tähevahed on erinevad jne. Vihjamine üritab seda parandada, korrigeerides tähemärgi kuju ja üritades tähte teha kuvatavaks ja esteetiliselt korrektseks võimalikult väikese resolutsiooni ja punktide arvu juures.

## Vihjamine šriftis

Vihjamine on šrifti süda. Selle loojad, tuginedes tähelepanekutele ja mitmesugustele arvamustele “korrektsest” vihjamisest, otsustasid, et ei ole olemas kindlat unikaalset vihjamise parameetrit, mida rakendada automaatselt kõigile tähemärkidele. Selle asemel sidusid nad lihtsa rasteriseerimisrakenduse uue programmeerimiskeelega. Iga tähemärk sisaldab oma isiklikku väikest vihjamise tarkvara, millega saab manipuleerida tähemärgi piirjoone kontrollpunkte vahetult enne rasteriseerimist, täpselt nii nagu vihjamise tarkvara soovib. Praktikas tähendab see, et iga tähemärgi piirjooned on hoolikalt painutatud, muudetud on ainult neid pikseleid, mida on vaja, saavutamaks perfektset pikselkujutist.



Joonis 12. 10 punktine New York šrifti “e” 300 dpi (punkti tolli kohta) resolutsiooni korral: esmalt rasterdatud ilma vihjamiseta, teise “e” puhul on näha piirjoone muutmise vastavalt vihjamisjuhiste ja viimane on rasterdus peale vihjamist – juba tunduvat korrektsem ja sarnasem piirjoonena salvestatud tähekuju.

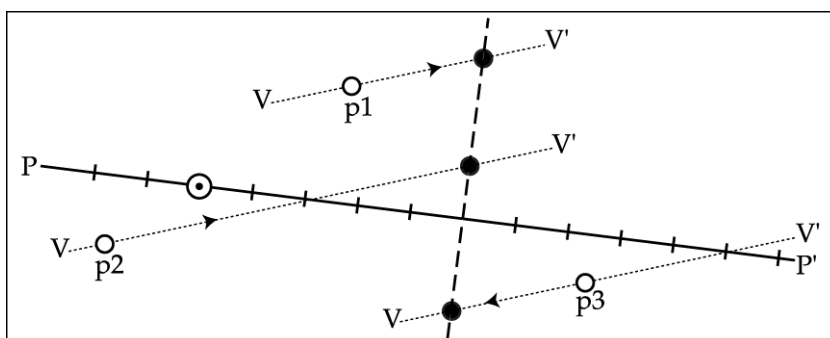
### Tähemärgi suuruse muutmine

Enne kui rakendatakse vihjamist, suurendatakse tähemärk vastava hetke ppem-idele. (Mõõtühik ppem (*pixels per em*) näitab rasterisatsiooniks vabade pikslite arvu, tuletatud tähemärgi punkti suurusest ja resolutsioonist, kusjuures em võrdub tähemärgi “m” laiusega.) Kõik tähemärgi koordinaadid on mõõdetud murdosa pikslitega, mitte šrifti loomisel kasutatud täisarvuliste ühikutega. Koordinaadid on 32-bitised korrastatud punktid täpsusega kuus kohta peale koma ühe biti kohta. Iga punkt talletatakse kahel korral: esmalt originaalse asetuse jaoks ja teiseks tema uue parandatud positsiooni jaoks.

### Punktide liigutamine ja projektsioon

On olemas juhendid absoluutseteks ja suhtelisteks punktide liigutamiseks, vastavalt parameetritele, mis on võetud kontrollväärtuste tabelist, või teis-

test punktidest, mis on juba liigutatud. Kogu punktide liigutamise protsess juhendub kahest vektorist: projektsiooni vektorist  $P$ , piki mida võetakse kõik ühikud ja "vaba" vektoris  $V$ , mida mööda muutused tegelikult toimuvad. TrueType tehnoloogia paindlikkus seisneb nende vektorite osalisel iseseisvusel ja võimaluses paigutada neid ka diagonaalselt.



Joonis 13. Kolmele punktile  $p_1$ ,  $p_2$  ja  $p_3$  antakse ühine väärtus projektsiooni vektoril  $P$ . Iga punkti liigutatakse piki tema "vaba" vektorit  $V$ , kuni lõikumiskohani projektsiooni-vektoril defineeritud kaugusega, seejärel defineeritakse sinna punkti uus asukoht.

## Šrifti ühtlustamine, kasutades kontrollväärtusi

Üks vihjamise võtmefunktsioon on säilitada šrifti "hinge" läbi kõigi tähemärkide. Sellel eesmärgil ongi loodud TT puhul kontrollväärtuste tabel KVT. See peab sisaldama ühikuid, tähemärkide sarnaste omaduste defineerimiseks, et neid siis ühtlustada terve tähestiku lõikes. KVT-tabeli väärtused on alati skaleeritud vastavalt hetkel kasutatavale resolutsioonile (*ppem*).

### Näidis KVT-tabeli kirjed võiksid olla järgnevad:

- Kui jämedad on tähe vertikaalsed detailid?
- Kui jämedad on tähe horisontaalsed detailid?
- Kui paksud on seriifid?
- Kui palju kalduvad ümarjooned kõrvale põhijoonest (algsest tähe kontuurist)?

Tihti on sobiv manipuleerida punktidega, mis ei ole lõplikul piirjoonel, nende defineerimiseks kasutatakse "hämar ala" ja nende väärtused talletatakse jällegi KVT-tabelisse. KVT-tabelis asuvad ka vihjamise funktsioonid. Igas programmeerimiskeeles muudavad funktsioonid või asendused programmi väiksemaks ja kergemini hallatavaks. TT võib funktsioone defineerida KVT-tabelis, aga mitte tähemärgi programmis. Enne kui tähemärgid rasteriseeritakse pikselpinnaks, käivitatakse KVT-tabelis asuvad funktsioonid.

## Delta-vihjamine

TT kood (kui ta on hästi kirjutatud) võimaldab šrifti suurendada ja vähendada väga erinevatele suurustele, parimad šriftid kasutavad selleks otstarbeks Delta-vihjamist, juhendit nihutamaks kontrollpunkte veidi erinevate suuruste puhul. Instruktsioon kõlab umbes järgnevalt: "12 pikslise kirja korral nihuta viiendat kontrollpunkti 1,25 pikslit", pöörates tähelepanu vektoritele  $P$  ja  $V$ . Samuti teeb Delta-juhend korrekture KVT-tabeli väärtustele, tavaliselt toimub see eelprogrammis ja kiirendab seega tööks vajalikku protsessi.

## “Ära jäämise” kontroll

Mõningate tähemärkide puhul on vihjamist äärmiselt raske teostada, eelkõige keerukate tähemärkide puhul, või juhul kui tähemärke on “keeratud”. Sellistel juhtudel on kasulik lülitada välja TT “ära jäämise” funktsioon, mis tähendab, et tööprotsessi kiirendamiseks ei üritata keerukaid tähemärke kuvada perfektselt.

### Esitlus

Te võite ette kujutada, et eelnevalt kirjeldatud vihjamise programmi jooksutamine tekstide kuvamiseks kulutab suure hulga riistvara ressursist ja kindlasti ei taha ju keegi, et šriftid arvuti aeglaseks teevad. TT vähendab riistvara koormust:

- **Kasutades vihjamise meeldejätmist** (ing *cache*), kui tähemärk on juba ükskord vihjatud, siis kasutatakse selle pikselkujutist korduvalt (nagu enamuse skaleeritavate šriftide puhul). Macintosh ning Windows süsteemid suurendavad ja vähendavad meeldejäetavate märkide arvu vastavalt vajadusele automaatselt.

- **Sageli kasutatud väärtuste eelarvutamine**, kasutatakse kõige tähtsamate väärtuste ette arvutamiseks nagu tähemärkide laius ja levinumad suurused.

## TrueType ja automaatne vihjamine

Tarkvara lahendused, mis väidetavalt konverteerivad Type 1 šrifte TT šriftideks, teevad sageli katseid kopeerimaks vihjamist TT šrifti. DTP Type Designer ja FontLab 3.0 tõlgivad vihjamist Type 1 šriftis. Fontographer pakub kolmel tasandil TT šrifti auto-vihjamist: väljajätmise vältimine, auto-vihjamine ja auto-vihjamine + diagonaalne kontroll. Aga ei ole üheselt mõistetav, mis rakendusi kasutatakse auto-vihjamise protsessis, niisiis parimad tulemused saavutab kolmekordse vihjamisega, mis sageli lõpeb paraku veateatega.

Probleeme on ka TT šriftide tööstuslikus tootmises, ülesanne kirjutada vihjamisrakendused igale tähele igas šriftis, mis sa lood, on väga ajamahukas (näiteks šriftiperekonnale, kus on 8 šrifti, tuleb kirjutada umbes 2000 pisikest programmi). Isegi ekspert TT vihjajad kasutavad vahel auto-vihjamist, kus nad siis hiljem käsitsi korrektureid teevad. Šrifti tarkvara StingRay on selleks sobilik, aga Fontographer'it või FontLab'i on võimalik kasutada ainult mõnedel juhtudel.

Auto-vihjamine saab olla efektiivne ainult juhul, kui šriftis on palju tähemärke, mida on vaja omavahel ühendada. Kui automaatseid rakendusi kasutada logo või graafilise kujundi juures, ei muutu see sugugi paremaks, kui vihjamata versioon.

### Materjalid

---

*How does TrueType hinting differ from other outline font formats?* – Microsoft - [www.microsoft.com/typography/hinting](http://www.microsoft.com/typography/hinting)

*Font Quality Specifications* – Apple - [developer.apple.com/fonts](http://developer.apple.com/fonts)

*TrueType Typography* - TYPE chimérique - [www.trueType.demon.co.uk](http://www.trueType.demon.co.uk)

*TrueType Hinting* - Microsoft - [www.microsoft.com/typography/hinting](http://www.microsoft.com/typography/hinting)

*Basic hinting philosophies and TrueType instructions* - Microsoft - [www.microsoft.com/typography/hinting](http://www.microsoft.com/typography/hinting)

*Get The Hint!* – Tom Rickner, Daniel Will-Harris - [www.will-harris.com](http://www.will-harris.com)

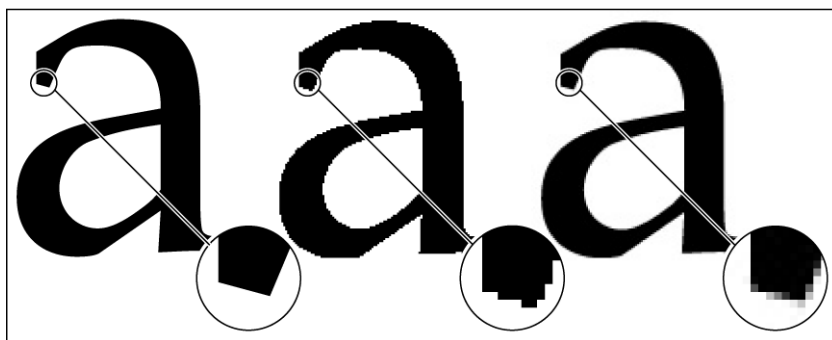
*The raster tragedy at low resolution* – Microsoft - [www.microsoft.com/typography](http://www.microsoft.com/typography)

## Šriftid ja “anti-aliasing”

### Mis on *anti-aliasing*?

*Aliasing* on tuntud efekt kuvaritel, tegelikult enamusel pikselriistvaral, kus kasutatakse diagonaalseid ja kõveraid vektoreid, kuvades neid tihe- da sik-sakina. Kui pikslid on suured, nagu vanematel kuvaritel, on ilmselge vajadus tehnoloogia järele, mis pehmemdaks graafikat. *Anti-aliasing*, on loomulikult tehnika, mis on loodud vähendamaks *aliasingu* efekti. Lisades täistoonides sik-sakile kriitilistesse kohtadesse pooltoone, tekitab rakendus mulje ühtlastest sujuvatest kõveratest. Lihtne teoreetiline mudel *anti-aliasingu* teostamiseks näeb ette, et kujund renderdatakse vastavalt konkreetsele lõplikule ekraaniresolutsioonile, ümardades kujundi kontuurid vajalike pooltoonideni. Aga tuleb välja, et see meetod ei ole tegelikult parim lahendus, on olemas korrektsemaid korrektureid tegemise filtreid.

*Anti-aliasing* efekt sõltub kuvarist, tähtis on kui palju värve kuvar suudab näidata: kuvar, mis näitab ainult 256 värvi, ei suuda teha korrektselt tööd piiratud pooltoonide arvu tõttu. Kui värve aga mitte arvestada, on asi tunduvalt lihtsam, monitor, mis võimaldab 16 musta ja valge vahelist pooltooni, on juba ideaalne teksti *anti-aliasingu* teostamiseks must-valgelt. Värviliste lahenduste puhul on vaja lihtsalt piisavalt üleminekutoone taustast kujundi või teksti põhitoonile.



Joonis 14. Vasakult vektorkujutis, aliasing efekt ja anti-aliasing korrektureid. Suurendustel on täpselt näha, mis toimub kujutise piirjoonega.

### *Anti-aliasing* ja tüpograafia

Kuvari-põhise tüpograafia puhul on oluline loetavus, samas on aga mõistetud, et *anti-aliasing* võib mõnel juhul teksti muuta raskemini loetavaks. Me ei mõista täielikult nägemise protsessi tähtede tajumisel (loe: lugemisel), aga mõned *anti-*

*aliasingu* kohandused on kindlasti parandanud mõlemat - teksti loetavust ja esteetilist korrektsust ekraanil.

IBM'i Yoda kuvar, väljatöötatud 1986-nda aasta paiku, toetab *anti-aliasing* teksti. Esimene personaalarvuti, mis toetas *anti-aliasing* rakendust tekstile oli Acorn Archimedes oma RISC OS operatsioonisüsteemiga.

Microsoft võrdleb tehnoloogiat, mida kasutati Windows 95's tekstide pehmendamiseks: vertikaalsed ja horisontaalsed jooned on jäetud puutumata, pehmendatud ja korrigeeritud on ainult kõveraaid, seriife ja diagonaale. *Anti-aliasing* teksti kutsutakse sageli "must-valge" (ing *grey-scale*) tekst.

handiworks handiworks

Joonis 15. Esimese sõna puhul on tegemist Adobe Photoshopi *anti-aliasing* efektiga. Pane tähele kuidas vertikaalsed ja horisontaalsed jooned on muutunud uduseks ja sherifid on peaaegu kadunud. Teine sõna, on genereeritud kasutades Windows 95'e *anti-aliasing* rakendust. Iga täht on renderdatud individuaalselt, Vertikaalsed ja horisontaalsed jooned on jäetud täis toonidele, pehmendatud on ainult kõveraaid, diagonaale ja sherife. (Tegemist on Times New Romaniga šriftiga, suurendus 4 korda)

## ***Anti-aliasing* ja Macintosh**

Üks rakendusi Macintosh arvutitele on SmoothType 2.0.2, mille on loonud Greg Landweber. See on suurepärase jaosvara (kopeerimist ja proovimist ei piirata, kuid pidevaks tööks nõutakse väikest registreerimistasu - 10\$). SmoothType 2.0.2 lisab *anti-aliasing* efekti kõigile TT šriftidele (samuti Type 1 šriftidele, kui ATM on installeeritud). Võimalus on valida kahe meetodi vahel: kiirem 2-bitine *anti-aliasing* (4 pooltooni) ja aeglasem 4-bitine (16 pooltooni).

Materjal

---

*TrueType Typography* - TYPE chimérique - [www.truefont.demon.co.uk](http://www.truefont.demon.co.uk)  
*Smooth Fonts in Windows 95* – Microsoft - [www.microsoft.com/typography](http://www.microsoft.com/typography)  
*Anti-Aliasing of fonts on the Acorn Archimedes, using the RISCos operating system* - by Matthew Godbolt. - [newton.ex.ac.uk/general](http://newton.ex.ac.uk/general)  
*The new font system from Type Solutions* - Type Solutions - [www.typesolutions.com](http://www.typesolutions.com)

## OpenType šrift

**TrueType Open** on tagajärg Microsofti reageeringust **TrueType GX**ile. Loodud eelkõige keerulisemate tüpograafiliste nõudmistega keelte e mitte ladinakeelkonna keelte tarvis, nagu näiteks araabia keel, samuti on TT Open vajalik programmidele ja töökeskkondadele, mis töötavad korruga rohkem, kui ühe keelega, see toetab alternatiivseid tähemärke ja mõtteseoslike täheühendite asendust. Microsoft avalikustas TrueType Open tehnoloogia 1995, napilt enne seda, kui turule tuldi operatsioonisüsteemiga Windows 95, mis kasutas juba selle rakendusi.

TT Open šrifti genereerimine on märgatavalt lihtsam kui GX šrifti oma, aga šrifti on kodeeritud ka tunduvalt vähem tüpograafilisi võimalusi kui GX'i, näiteks ei suuda TT Open teostada mitme omavahel vaheliti asetseva šrifti teisendust.

TrueType Open'ist arenes välja **OpenType**, ühendav lüli Microsoft'i ja Adobe vahel, mis viis kokku TrueType ja PostScript Type 1 šriftid. Alates sellest hetkest sisaldab Type 1 Multiple Master tehnoloogiat, OpenType võttis üle ka paljud TrueType GX'i omadused - näiteks kasutab ta GX'i tähtede liitmise rakendust.

### OpenType šrifti põhilisemad omadused on:

- OpenType šrift lubab mitmesugust vastastikkust konvertimist nii üksikute kirjamärkide, kui ka tähtemärkide vahel, toetab täheühendeid ja liittähti, kirjamärkide positsioonilisi paigutusi, alternatiivseid kirjamärke ja muid asendusi.
- OpenType šrift sisaldab infot võimaldamaks tähemärkide positsioneerimist ja kasutamist kahemõõtmelises töökeskkonnas.
- OpenType šriftis on üksikasjalik kirjeldus keelelise informatsiooni kohta, seega saab tekstitööstlustarkvara kohandada seda vastavalt vajadustele ja kasutaja nõudmistele.
- OpenType šrift on avatud formaat, võimaldades kasutajatel lisada täiendusi ja tüpograafilisi omadusi.

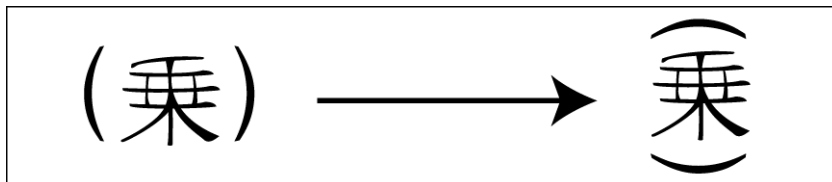
## Pilguheit OpenType omadustele

OpenType šrift võib sisaldada mitmeid alternatiivseid tähemärke ühe tähe sümboliseerimiseks ja mehhanismi, millega on võimalik neid vastavalt vajadusele kasutada. Näiteks araabia keeles on tähe kuju sageli erinev, olenevalt sellest, kus kohal täht sõnas asub. Nagu näidatud joonisel 16, täht *ha* võib olla ükskõik milline neljast, olenevalt sellest, kas ta asub iseseisvalt, seisab sõna alguses, keskel või lõpus. OpenType tehnoloogia aitab tekstitööstlustarkvaral tuvastada, millist varianti oleks tekstis vaja parajasti kasutada.



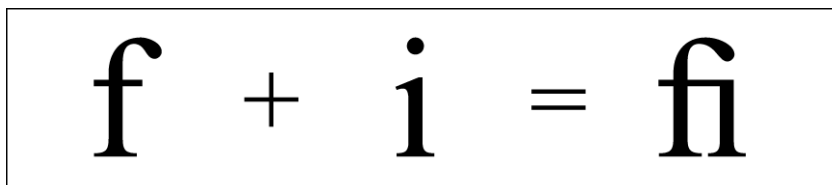
Joonis 16. Üksik, sõna alguses, keskel ja lõpus paiknev araabia tähestiku täht ha.

Sarnaselt aitab OpenType šrift tarkvaral kasutada ka korrektset paigutust kirjamärgist/märkidest, kui näiteks tekst peaks olema paigutatud horisontaalselt vertikaalse asemel, vaata joonist 17. Kanji kasutab alternatiivset paigutust ümarsulgude puhul.

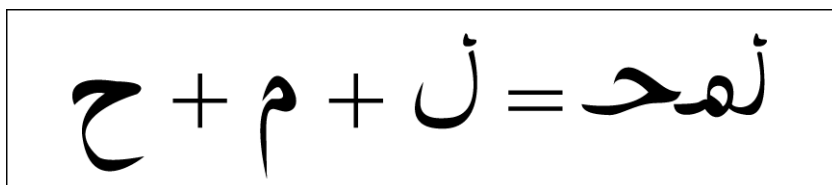


Joonis 17. Alternatiivne ümarsulgude paigutus kanji puhul, kasutatakse automaatselt kui trikkida kanji vertikaalselt paiknevate sulgude vahele.

OpenType šrift toetab samuti täheühendite moodustamist ja vastupidi. Näiteks inglise, prantsuse ja teised ladina tähestikul baseeruvad keeled võivad asendada täheühendi nagu "fi", tema komponentide, tähemärkidega - antud juhul "f" ja "i". Vastupidiselt, üksikud "f" ja "i" tähemärgid võib asendada täheühendiga, võimalus on anda tekstitöötlusprogrammile rohkem paindlikust, paigutamaks täheühendeid rea täitmisel või teksti joondamisel.



Joonis 18. Kaks ladina tähte ja nende täheühend.



Joonis 19. Kolm araabia tähemärki ja neist moodustuv täheühend.

Tähemärkide asendus on üks uuendusi OpenType mitmetahulises tehnoloogias. Tähemärkide paigutamiseks kasutatakse täpseid X ja Y telje koordinaate, OpenType šrift võib liita kirjamärkide punkte, sidumaks märke omavahel, genereerimaks kursiivis teksti ja tähemärke, mis vajavad rõhu või muid spetsiaalseid märke.

OpenType šrift võib sisaldada ka tekstirea infot, mis määrab ära, kuidas paigutada kirjamärgid horisontaalselt või vertikaalselt. Kuna tekstirida võib erineda erinevates programmides (kirjavahemärkide kogumikes), siis see info on eriti vajalik teksti joondamiseks, milles on kasutatud rohkem, kui ühe tähestiku või šrifti tähemärke.

# Tartu 74 سیبوزم

Joonis 20. Tekstirea järgi joendus, kombineeritud on ladina ja araabia tähestikku.

## TrueType vs OpenType

TT šrift on kolleksioon erinevatest tabelitest, mis sisaldavad mitmesugust infot: tähemärkide piirjooni, proportsioone, informatsiooni paiknemise kohta tekstireal jne. OpenType sisaldab kõiki neid omadusi, mida TT, lisaks veel täiendavaid tabeleid spetsiifilisema tüpograafilise informatsiooniga.

Tekstitötlustarkvara/programmid - vahel nimetud ka OpenType'i "klientideks" - suudavad leida ja analüüsida informatsiooni OpenType šrifti tabelites. Näiteks teksti töötlev klient võib valida korrektsed tähekujud ja paigutada nad korralikult tööse.

Võimalusel defineerivad OpenType šrifti tabelid ainult infot, mis puudutab šrifti paigutust. Tabelid ei ürita kodeerida teavet mingi kindla keele või kindla šrifti tüpograafiliste omaduste kohta. Selline informatsioon, mis puudutab kõiki šrifte mingis kindlas keeles, antakse tekstile tekstitötlustarkvara poolt, mitte ei talletata seda šriftis.

## Šrifti terminoloogia

OpenType šrifti mudel keskendub järgnevate märgusõnade ümber: tähemärgid, tähestik, keele süsteemid ja omadused.

### Kirjamärgid vs tähemärgid

Kasutajad ei näe ega prindi kirjamärke: kasutaja näeb ja prindib tähemärke. Tähemärk on kirjamärgi esitus. Kirjamärk "suurtäht A" tähistatakse tähemärgiga "A" nii Times New Roman Bold šriftis kui ka Arial Bold šriftis. Šrift on kolleksioon tähemärke. Selleks et leida/kuvada/printida tähemärki, kasutab klient šrifti tabeleid, mis omakorda väljastavad vastavalt kliendi poolt sisestatud kirjamärgi koodile sellele vastava tähemärgi (tähemärkide tabelist).

Tähemärk võib olla ka kombinatsioon kirjamärkidest või alternatiivsetest kirjamärkidest: tähemärgid ja kirjamärgid ei vastandu üks-ühele. Näiteks kasutaja võib sisestada kaks kirjamärki, mida saab esitada paremini ühe tähemärgina - liittähena. (vt joonis 18) Sama täht võib olla ka erinev, olenevalt sellest, kas ta asub iseseisvalt, seisab sõna alguses, keskel või lõpus (vt joonis 16), niisiis on šriftil vaja mitut erinevat tähemärki, et esitada üht ja sedasama kirjamärki. OpenType šriftid sisaldavad tabeleid, mis pakuvad klientidele infot võimalike tähemärkide asenduste kohta.



Joonis 21. Mitmed tähemärgid kirjamärgi "ja" tähistamiseks.

## Tähestik

Tähestik on koostatud grupist ühenduses olevatest kirjamärkidest, mida võib kasutada ühes või mitmes keeles. Ladina, araabia ja thai tähestik on näidised toodud joonisel 22. Šrift võib kasutada ühte või mitut tähestiku. OpenType šrifti sees on tähestikud identifitseeritud unikaalse 4-baidise märkega.



Joonis 22. Tähemärgid ladina, kanji ja araabia tähestikust.

## Keele süsteemid

Tähestiku võib jagada omakorda mitmeks keele süsteemiks. Näiteks kasutavad ladina tähestikku kirjutamiseks nii inglased, prantslased kui sakslased, aga ometi on igal keelel oma spetsiifilised nõudmised tekstitöötusele. Šrifti loojad võivad valida, kas nad keskenduvad tähestikule või keele süsteemile või mõlemale.

Keele süsteem, erinevalt tähestikust, ei ole tingimata nähtav, kui teksti töötuse klient uurib kasutatavaid kirjamärke. Et vältida ebaselgust, peab kasutaja või operatsioonisüsteem tuvastama šrifti keele süsteemi. Vastasel juhul kasutab klient vaikimisi määratud keele süsteemi, mis on iga tähestikuga kaasas.

A charming mess  
Le cahier français  
Das Wasser war heiß

Joons 23. Erinevused inglise, prantsuse ja saksa keele süsteemis.

## Omadused

Omadused defineerivad šrifti põhilisi funktsioone. Šrift, mis sisaldab tabeleid diakriitiliste märkidega, on märgistatud tavaliselt märkega "mark" omadus. Šrift, mis toetab vertikaalsete tähemärkide asendust kannab märki "vert" omadus.

OpenType šrifti omaduste mudel kindlustab suure paindlikkuse šriftide loojate tarvis, kuna omadused ei pea olema eelnevalt defineeritud Microsoft'i poolt. Hoopis vastupidi, šriftide loojad saavad töötada koos tarkvara arendajatega, loomaks vajalikke uuendusi šriftidele, lisades sellised uuendused OpenType šriftidele ja võimaldades kliendi tarkvaral neid toetada.

## OpenType šriftid ja Windows 95

Windows 95 operatsioonisüsteemi baasšriftid Kesk-Ida ja Kaug-Ida versioonides on OpenType šriftid. Need šriftid demonstreerivad edukalt OpenType tehnoloogia mitmekülgust

### Kesk-Ida Windows 95

Kesk-Ida Windows 95 kasutab mitmeid araabia tähestikuga OpenType šrifte: korrigeeritud standardjämedusega, proportsionaalse standardjämedusega, parandatud paksude (*bold*) tähemärkidega ja proportsionaalsete paksude tähemärkidega. Need šriftid kasutavad mitmeid OpenType tehnoloogias olevaid tähemärkide asendamise rakendusi, lihtsaid asendusi (üks-ühele sõltuvalt kontekstist), tähemärkide asendamist liittähtedeks (mitu-üheks) ja lisaks kasutaja defineeritud asendused. Kesk-Ida versiooni puhul tegeleb tähemärkide asendusega operatsioonisüsteem ise, kasutades iga šrifti puhul tema GSUB-tabelit.

### Kaug-Ida Windows 95

Kaug-Ida Windows 95 kasutab samuti mitmeid OpenType šrifte: korrigeeritud seriif, proportsionaalne seriif, korrigeeritud sans seriif ja proportsionaalse sans seriif kogumikku. Jaapani šriftid kasutavad laialdaselt OpenType šriftide lisaomadusi, kaasa arvatud vertikaalset tähemärkide asendust ja baasjoonest lähtuvat joondust. Sarnaselt Kesk-Ida versiooniga, tegeleb tähtede asendusega siingi operatsioonisüsteem ise, kasutades jällegi GSUB-tabelit. Baasjoone järgi joondamist peab aga haldama teksti-töötlustarkvara, kasutades selleks iga konkreetse šrifti BASE-tabelit.

Open Type tabelleid on täiendavalt vaadeldud lisis 2.

#### *Materjalid*

---

*OpenType specification* - Microsoft Corporation - [www.microsoft.com/typography](http://www.microsoft.com/typography)  
*OpenType initiative FAQ* - Microsoft Corporation - [www.microsoft.com](http://www.microsoft.com)  
*Microsoft and Adobe Systems to Deliver Universal Font Format* - Microsoft Corporation - [www.microsoft.com](http://www.microsoft.com)  
*Agfa OpenType white paper* - Agfa-Gevaert Group - [www.agfahome.com](http://www.agfahome.com)  
*OpenType / TrueType Open* - TYPE chimérique - [www.truetype.demon.co.uk](http://www.truetype.demon.co.uk)

Üheksas peatiikk. Kuna ekraanigraafikast ei saa tulevikus ilmselt ei üle ega ümber, vaatlen ka Microsoft'i viimast saavutust ClearType tehnoloogiat.

Microsoft'i suurepäraseimal saavutusel Windows XP juures ei pruugi olla midagi tegemist tema esiletrüüivate paljukiidetud uuendustega. Revolutsiooniline on hoopis see, et uus operatsioonisüsteem kasutab Microsoft'i ClearType šrifti renderdamistehnoloogiat, mis muudab tekstid vedelkristall-ekraanidel (LCD- Liquid Crystal Displays) uskumatult selgeks. Võib kirjeldada erinevust tavalise renderduse ja ClearType tehnoloogia vahel ühe sõnaga – shokeeriv. Kineskoopmonitoridel ei muuda tehnoloogia midagi. Aga inimestele, kes kasutavad kaasaskantavaid arvuteid või lameekraani, võib Windows XP muljetavaldavalt tõestada, kuidas ta parandab teksti kvaliteeti, iseäranis dokumentide ja interneti lehtede puhul, kus must tekst on valgel taustal.

Joe Wilcox - c | net - 24 June 2001

## ClearType

ClearType on Microsoft'i loodud tarkvaraline lahendus, parandamaks tekstide loetavust populaarsetel LCD ekraanidel. LCD tehnoloogiat kasutavad laialdaselt *laptop*-arvutid, Pocket PC'd ja lamedad õhukesed kuvarid. ClearType šriftis tekst tundub ekraanil peaaegu sama terav ja selge, kui väljatrükituna paberil.

ClearType kasutab oma rakenduse teostamiseks LCD ekraani iga piksli vertikaalsete värvitriipude elemente. Enne ClearType tehnoloogiat oli väikseimaks detailide kujutamise ühikuks ekraanil üks piksel, aga ClearType, joostes LCD ekraanil, kasutab teksti detailide kuvamiseks murdosa pikslit horisontaalselt. Lisaresolutsioon tõstab ekraani teravust ja toob välja peeneid detaile teksti puhul, tehes pikaajalise arvutiteksti lugemise tunduvalt vähem vastumeelseks.

ClearType on Microsoft'i järjepideva šriftiarenduse etapivõit. Rohkem kui kaks aastat otsis Microsoft'i töörühm lahendust, et parandada šriftide renderdamise protsessi, detailselt uuriti tüpograafiat ja lugemise protsessi. Inimesed suudavad süveneda lugemisse jäägitult vaid juhul, kui sõnade äratundmine toimub alateadlikult ja teadvus on vaba jälgimaks teksti mõtet.

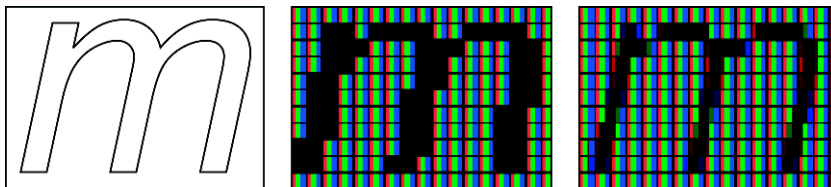
Avastati, et alateadlik sõnade äratundmine on võimalik ainult siis, kui tähemärkide tüpograafilised omadused, nagu kuju ja paksus ning tähtede omavaheline kaugus, on tasakaalus, sõnad moodustavad sellistel tingimustel kergesti haaratava mustri. Nende teadmistega, asus Microsoft lähemalt vaatlema, kuidas teksti ekraanile renderdatakse.

## ClearType tehnoloogia

Selleks, et mõista kuidas ClearType töötab, tuleb esmalt teha endale selgeks, mis vahe on LCD monitoril võrreldes teiste tehnoloogiatega. Enamus kuvarid kujutab pilti kasutades piksleid, mis suurendatult tunduvad üksiku ruuduna. Seevastu üks piksel LCD ekraanil moodustub kolmest alam-pikslit: punasest, rohelisest ja sinisest (*red - green - blue* e RGB). Inimese silmaga vaadatuna moodustab see alam-pikslite kolmik ühe piksli.

Kuidas see mõjutab digitaalsete šriftide kvaliteeti? Traditsiooniline arvutišrift eeldab, et piksel on kas "sees" või "väljas", vastavalt, kas ilmudes ekraanile või mitte. Tähed ilmuvad vastavalt pikslite olemasolule ekraanile sakiliste servadega, kuna nad koosnevad väikestest kastidest (pikslitest). Tavaline *anti-alias* efekt (vaadeldud eespool) pehmen­dab sik-saki, muutes osad pikslid pooltoonideks, samas kaotab aga ka šrifti teravus.

ClearType tehnoloogia teab, et LCD ekraani pikslid koosnevad värvilistest alam­pikslitest. ClearType kasutab inimese nägemisest lähtuvat mudelit, valimaks alam­pikslitele heleduse astet. ClearType tähed ilmuvad tänu sellele ekraanile pehmelt, mitte sakilistena ja ometi on tähemärkide servad teravad.



Joonis 24. ClearType šrifti renderdus

1. Väiketäht "m" tähemärgi piirjoonena, nii nagu ta on salvestatud šrifti.
2. Suurendatud "m" renderdus ekraanil ilma ClearType rakendusteta. Pane tähele, kui sakilised on tähemärgi servad.
3. Suurendatud "m" renderdus ekraanil, kasutades ClearType tehnoloogiat, tähemärk on tunduvalt pehmem, kui eelmisel renderdusel.

ClearType šrifti tehnoloogia töötab kõigi kuvaritega, loetavus paraneb mingil määral ka kineskoopkuvaritel (CRT), aga mitte nii drastiliselt, kui värvilistel LCD monitoridel või kõrgkvaliteediga laptop arvuti ekraanide puhul.

Hetkel on Windows XP Microsoft'i esimene operatsioonisüsteem, mis toetab ClearType rakendusi süsteemipõhiselt.

*Materjalid*

*Microsoft ClearType - Microsoft Corporation - [www.microsoft.com/typography/cleartype](http://www.microsoft.com/typography/cleartype)*

*Win XP: To upgrade or not to upgrade? - Joe Wilcox - [news.com.com](http://news.com.com)*

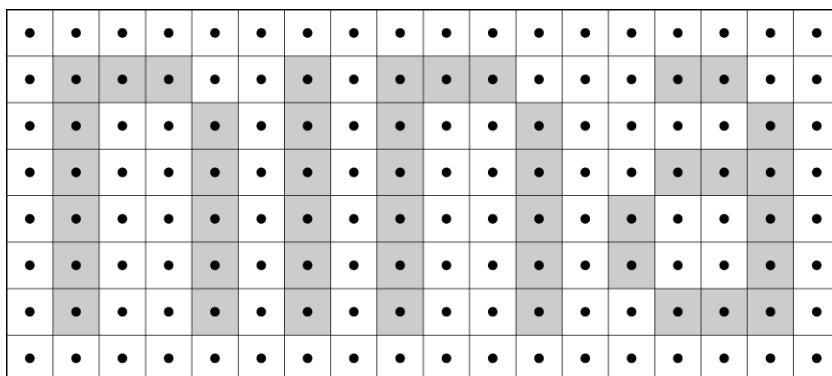
Kümnes peatükk. Selleks, et šriftid oleksid ekraanil korrektsed, ei ole ainuke võimalus neid kuidagi udutada või filtreerida (*anti-alias*, *ClearType*), alternatiivne ja aina laialdasemalt leviv võimalus on pikselšriftid.

## Pikselšriftid

Šriftid, mis on disainitud trükiste ja kõrge resolutsiooniga seadmete tarvis, ei pruugi väikestena arvuti ekraanil olla korrektsed. Mida väiksem on kiri, seda hullemaks läheb tavaliselt väljanägemine. Probleem polegi nii oluline, kui tegemist on mingi etapiga trükiste ettevalmistamises või muus disainis, aga oluliseks muutub väikeste tekstide loetavus juhul, kui tegemist on ainult kuvarilt vaatamiseks mõeldud toodetega - internetilehed, interaktiivsed CD, visuaalsed tarkvaralahendused jms.

Tähemärgid, mille pikslid ei ühti oma kuju tõttu ekraanisagedusega, on ebakorrektsed seetõttu, et nad on kompromisslahendus ekraani resolutsiooni ja šrifti pikslite vahel. Kui väikeste šriftide puhul kasutatakse ka *anti-alias* efekti, vähendamaks sakilisust, muutub tekst uduseks ja raskesti haaratavaks.

Pikselšriftid on lahendus, disainitud spetsiaalselt ekraani resolutsiooni arvestades, iga joon ja punkt langeb perfektselt kokku ekraani tiheda ruudustikuga. Isegi väga väikestel suurustel on šriftid selged ja puhtad. Pikselšriftid väldivad enamasti kõveraid ja kaari, eelistades sellele igal võimalusel horisontaalseid ja vertikaalseid jooni.



Joonis 25. Nina on Microsoft'i sans seriif tähestik, disainitud vastama ekraani punktidele, edukalt kasutatav väikeste tekstide esitamiseks ekraanil. Šrifti on loonud Matthew Carter, ja käsitsi on vihjamise lisanud Tom Rickner.

Pikselšriftide puuduseks võib lugeda, et nad on fikseeritud suurusega st nende suurust ei ole mõttekas muuta. Erandiks on, kui suurust muuta kordades, siis on šrift lihtsalt topelt paksem e kahekordse suurenduse korral on originaalis kavandatud ühe piksli asemel neli. Kui šrift on suurendatud arvestamata originaalsuurust, muutuvad tähemärgid ebahühtlaselt paksemaks ja on sageli lausa inetud.

### Ekraaniresolutsioonid erinevatel süsteemidel

Seostes ekraaniresolutsiooni ja šrifti suuruse vahel valitseb sageli suur segadus. Tavaliselt antakse teksti suurus punktides (*ing points*) – üks punkt on tüpograafiline mõõtühik e  $1/72$  tollist (toll on umbes 2,4 cm).

Alates turuletulekust 1984, on Macintosh arvutite ekraanide suhe järgmine - 1punkt = 1 piksel, nii lihtne see ongi. Macintosh arvutitel on alati olnud kõrge resolutsiooniga ekraanid ja pole olnud probleeme kuvada graafikat 72 pikslit tolli kohta.

Seevastu vanemad kontoriarvutid (PC'd) kasutasid aga viletsamaid kuvareid, suuremate pikslitega. Algselt kasutatud maatriks- ja nõelprinterite puhul ei olnud ka suurt vahet, kuna tööl, mida sa nägid ekraanil ja sellel, mis tuli välja printerist, ei olnud midagi ühist. Probleeme tekkis, kui laialdasele kasutusele jõudsid laser- ja tindiprinterid, PC'del olid aga endiselt suhteliselt viletsad monitorid. Seega, et vähem koormata arvutikasutaja silmi, muutis Microsoft Windows'i automaatselt ekraani resolutsiooniks 96 pikslit tolli kohta, efektiivsed 33% rohkem, kui tegelik väljatrüki suurus. Nad ei suutnud teha pikslit väiksemaks, nii siis tegid nad tolli pikemaks.



Joonis 26. Valik Atomic Media pikselšrifte, disainitud kasutamiseks nii veebis, Macromedia Flash rakendustes, kui ka WAP'ile loodud lahendustes.

Täna jagavad nii Mac, kui PC samasid kõrgeresolutsioonilisi kuvareid, aga sellele vaatamata on meid pandud olukorda, kus üks punkt Macintosh arvutitel on 1/72 tolli, seevast PC'l on ta pigem 1/54 tolli – ühe kolmandiku võrra suurem.

Hea uudis on, et veebi graafika puhul pole meil vaja muretseda punkti suuruse pärast. Arvuti ekraan, vaatamata arvutisüsteemi tüübile, on kindla arvu pikslitega nii horisontaalselt kui vertikaalselt. Kui on tegemist näiteks 15" monitoriga, on sellel enamasti resolutsioon 800x600 pikslit, 17" monitoril on resolutsioon pisut suurem, 1024x768 jne. Kui nüüd jagada ekraanil horisontaalselt paiknevate pikslite arv ekraani laiussega tollides, saame ppi (*pixels per inch*) väärtuse. Tuleb mees pidada, et ekraani suurus, mille annab tehas, on mõõdetud piki diagonaali ja mõõdetud on nähtavat ala. 17" monitori tegelik laius on seega umbes 12". Ppi väärtus ei kõigu erinevate kasutajate lõikes eriti, see on umbes 80-85 ppi (800/9.5 või 1024/12), võimaldades mugavat tekstide lugemist kvaliteetsel monitoril.

Tavalises tekstitöötlustarkvaras on teksti suurus defineeritud punktides, aga meid huvitab kõrgus pikslites, millel võib olla mingi seos ekraaniresolutsiooniga, aga samas ei pruugi seda ka olla. Kasutades pikselšrifte, tuleks igatahes teksti mõõtühikuteks panna pikslid, mitte punktid. Kui panna pikselšrift dokumenti, mis on tihedusega 72 ppi, on

pikslid võrdsed punktidega. Kui aga üritada panna šrifid näiteks 96 ppi tihedusega dokumenti, tekib probleem, kuna vaid vähesed tarkvaralised lahendused lubavad sisestada teksti puhul suuruseks 7.5 vms, et saavutada 10 piksline kiri. Et asju mitte keeruliseks ajada, tuleks oma dokumendi resolutsioon hoida alati 72 ppi. Ei ole mõtet töötada kasutades mõõtühikutena tolle, sentimeetreid, punkte või üldse reaalse maailma ühikuid, ainuõige on kasutada piksleid.

Enamuse tarkvarast, mis on mõeldud internetilahenduste loomiseks, on võimalik panna tööle, kasutades mõõtühikutena piksleid resolutsioonil 72 ppi. Programmid, nagu Adobe Photoshop ja Macromedia Fireworks, pakuvad tekstide korrigeerimiseks-pehmemdamiseks, sujuvamaks muutmiseks, mitmeid lahendusi, aga pikselšrifte kasutades tuleks need (*anti-alias* efektid) täielikult välja lülitada. Tekst tuleks sisestada parempoolse joondusega ja ilma tähevahede täiendava korrigeerimisega, reavahet võib suurendada vastavalt vajadusele. Keskele joondus võib põhjustada mõningast tähemärkide deformatsiooni, joondusel tehtavate ümardusvigade tõttu. Keskele joondades tuleks tähed hiljem käsitsi korrigeerida, tähelepanu tuleks pöörata tähevahedele.

### Kokkuvõtteks,

pikselšrifid õigustavad ennast järgnevatel tingimustel:

- šrifte on kasutatud samal suurusel, kui nad olid disainitud (suurenduste korral on seda tehtud kordades);
- šrifte ei ole kasutatud koos *anti-alias* efektiga;
- šrifte ei ole joondatud, rikkudes tähemärkide originaalseid vahesid.

*Materjalid*

---

*Pixel Fonts* - Joe Gillespie - [www.wpdfd.com/pixelfonts.htm](http://www.wpdfd.com/pixelfonts.htm)

*Let's Get Small* - Atomic Media - [atomicmedia.net](http://atomicmedia.net)

# KOKKUVÕTE

PostScript ja TrueType šriftide vahel on mitmeid erinevusi – samuti suhtelisi eeliseid nii ühel kui teisel, mis on aga tihti liialdatud. Mõlemad formaadid ja kombinatsioon nendest – OpenType – omavad uuendusi ja omadusi, mida ei ole endale teadvustanud enamuse kasutajaid. Praktikas kasutavad paljud mõlemaid formaate ja kombineerivad neid, vaatamata jutule suurtest erinevustest.

Siiski on olukordi, kus üks formaat oleks teisest parem, näiteks kui on vaja iseäralikke ekspertparameetreid (enamasti olemas PostScript šriftides või OpenType formaadis). TrueType šriftid ei tööta jälle mõnede vanemate tarkvaralahenduste puhul jne.

Šriftid, mis on disainitud trükiste ja kõrge resolutsiooniga seadmete tarvis, ei pruugi aga sobida jällegi ainult kuvarilt vaatamiseks mõeldud toodetele - internetilehed, interaktiivsed CD ja visuaalsed tarkvaralahendused. Lahendus on pikselšriftid, disainitud spetsiaalselt ekraani resolutsiooni arvestades, iga joon ja punkt langeb perfektselt kokku ekraani tiheda ruudustikuga. Isegi väga väikestel suurusel on šriftid selged ja puhtad.

Seevastu ClearType šrifti tehnoloogiat tuleks kasutada kindlasti värviliste LCD monitoride või kõrgkvaliteediga laptop arvuti ekraanide korral. ClearType kasutab oma rakenduse teostamiseks LCD ekraani iga piksli vertikaalsete värvitriipude elemente. Enne ClearType tehnoloogiat oli väikseimaks detailide kujutamise ühikuks ekraanil üks piksel, aga ClearType, joostes LCD ekraanil, kasutab teksti detailide kuvamiseks murdosa pikslit horisontaalselt. Lisaresolutsioon tõstab ekraani teravust ja toob välja peeneid detaile tähemärkides, tehes pikaajalise arvutiteksti lugemise vähem vastumeelseks.

Töös on käsitletud šrifti kontuure ja vektorloomust. Vektorgraafika võimaldab esitada tähemärke nii suuremalt kui väiksemalt, säilitades kujutiste sujuvad piirjooned. Vektorite eeliseks on, et neid on lihtne salvestada – infomaht on võrreldes pikselpindadega väike. Kõik vektorid ja kuvamiseks vajalikud punktid on määratletud koordinaatide ja kindlate valemitega. Matemaatiliselt korrektne pikslite esitus ei pruugi aga olla korrektne madala ekraani lahtusega kuvarite juures, tekst võib muutuda suisa loetamatuks. Vihjamise protsess on käsustik, koondamaks tehnikaid, mis peavad tagama teksti loetavuse ja maitsekuse ekraanil kuvamiseks. Teine protsess kvaliteedi parandamiseks on *anti-aliasing*.

Käsitlen digitaalset tüpograafiat alates 80-date keskelt kuni käesoleva hertkeni. Arvan, et olen andnud oma töös ülevaate suurematest tänaseni kasutusel olevatest šriftipõlvkondadest, samuti nende ajaloost ja kujunemisest. Detailsemalt saaks kindlasti peatuda formaatide spetsifikatsioonidel, mis täismahus küündivad sadade lehekülgedeni, pidasin seda siiski käesoleva töö jaoks liiga spetsiifiliseks.

Digitaalne tüpograafia on arenev valdkond, seega saab antud teemat kindlasti soovi korral täiendada ka tulevikus. Loodetavasti annab töö siiski piisavalt infot mõistmaks olnut ja põhimõtteid, millest hetkel lähtutakse.

Lisa 1. Täiendavalt töös kasutatud mõistete definitsioonid, toon ära ka kasutatud terminite ingliskeelsed vasted, vastavalt sellele, kuidas neid tõlkisin.

## Mõisted ja terminid

**Šrift** (ing *font*) - kogum sümboleid või tähemärke. Digitaalse teksti maailmas on šrift tähemärkide kogum või digitaalne info selle konstrueerimiseks.

**Tähestik** (ing *script*) - kasutatud peamiselt siis, kui on juttu mingist konkreetsest tähemärkide kogust, näiteks ladina tähestik.

**Tähemärk** (ing *glyph*) - konkreetne üks märk (antud töös peamiselt see, mida näeme ekraanil või väljaprintituna), näiteks suurtäht "A" šriftis Palatino .

**Kirjamärk** (ing *character*) - arvuti klaviatuurilt sisestatud märk näiteks "a". Ühele kirjamärgile võib vastata mitu tähemärki, näiteks võib "a" olla ekraanil nii suur kui väike "a", ka šrift võib olla näiteks Arial või Times.

**Spetsifikatsioon** (ing *specification*) - detailne funktsioonide kirjeldus, vajalik tarkvaraarendajatele ja šriftide disaineritele.

**Piirjoon, kontuur** (ing *outline*) – tähemärgi piirjoon kontuurina.

**Piksel- e rasterpind** (ing *bitmap*) - igale kujutise andmeelemendile/punktile vastab bitt, või bittide kogum.

**Punkt** (ing *point*) – üks punkt on tüpograafiline mõõtühik e 1/72 tollist.

**Skaleeritav** (ing *scalable*) – muudetava suurusega.

**Rasteriseerimine** (ing *rasterizing*) - protsess, mille käigus šrifti kontuur konverteeritakse pikselpinnaks, vastavalt väljundseadme vajadustele/parameetritele.

**Multi-platvorm** (ing *multi-platform*) - multi-platvormiline šrift e šrift, mis on kasutatav paljudel erinevatel arvutisüsteemidel, näiteks Mac ja PC.

**Kontuur-šrift** (ing *outline font*) - šrift, mida kirjeldatakse tähemärgi piirjoonte järgi, enamasti kasutatakse piirjoonte infot ka tähemärkide salvestamiseks.

**Vihjamine** (ing *hinting*) - šrifti või rakendustarkvarasse programmeeritud kogumik infot, selleks et korrastada/parandada šrifti *layout* i ja tähemärkide detaile.

Lisa 2. OpenType sisaldab täiendavalt viit uut tabelit: GSUB, GPOS, BASE, JSTF ja GDEF. Neid tabeleid ja nende sisu vaatlen järgnevalt lühidalt.

## OpenType tabelid

- GSUB:** Sisaldab infot tähemärkide asenduse kohta, et tegeleda juba detailsemalt iga tähemärgiga eraldi, üks-mitmeks asenduse (liittähtede lahutamise), mitmete tähemärkide asenduse (liittähtede) ja kaastekstist tuleneva tähemärkide asenduse kohta.
- GPOS:** Sisaldab infot tähemärgi paiknemise kohta X ja Y telje suhtes, et tegeleda iga üksiku tähemärgi joondamisega eraldi, infot paaris tähemärkide kohta, kursiivi ja kaastekstist sõltuva tähemärgi joondamise kohta.
- BASE:** Sisaldab infot põhiliste sarnasuste kohta tähestik-tähestikuks teisendamise jaoks.
- JSTF:** Sisaldab joondamise juhtnöore, kaasa arvatud sõnavahed ja Kashida joondus.
- GDEF:** Sisaldab informatsiooni iga üksiku tähemärgi kohta selles šriftis: täht (lihtne tähemärk, liittäht või kirjavahemärk), liitmispunktid (kui neid on) ja liittähe juurdelisamismärk (kui tegemist on tähemärkide liitmisega).

### Töötamine OpenType šriftiga

Tekstitöötlemise klient järgib standardset protsessi konverteerides kirjamärgi, mille kasutaja on sisestanud, positsioneeritud tähemärgiks, et tekitada teksti OpenType šriftiga:

- Kasutades šrifti CMAP-tabelit, korrigeerib klient kirjamärgi koodi tähemärgi koodiks.
- Kasutades infot GSUB-tabelist, korrigeerib klient tähemärgi koodi asendades positsioneerimise, vertikaalsed tähemärgid või liittähted.
- Kasutades GPOS-tabelit tähemärgi positsioneerimiseks ja tekstireala infot BASE-tabelist, paigutab klient tähemärgid kasutaja dokumenti.
- Kasutades JSTF-tabelit, joondab klient tekstiread, toimub ainult juhul, kui kasutaja on joondamise defineerinud.
- Operatsioonisüsteem rasteriseerib tähemärgid ja renderdab need vastavalt kasutatavate seadmete parameetritele. Eeldustekohaselt on need vastavuses väljundseadmetega.

Läbi eelnenud protsessi hoiab tekstitöötlus klient assotsiatsioone kirjamärkide originaalkoodide ja lõplikus töös asuvate renderdatud tähemärkide vahel. Lisaks võib klient salvestada keele- ja programmirakendusi teksti jadasse, et selgelt siduda tähekoodid tüpograafiliste rakendustega.