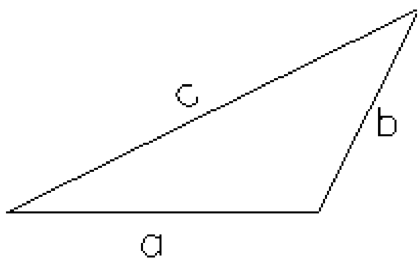
9 Grafiikka

Tässä luvussa käsitellään geometriaa ja graafisia kohteita. Mukana on pääosin alkeisoperaatioita.

9.1 Kolmio

Seuraavana tutkimme kolmiota:



Minkä tahansa kolmion ala saadaan kaavasta:

$A = [s(s-a)(s-b)(s-c)]^{1/2}$, jossa
a, b ja c ovat kolmion sivujen pituudet
s = kolmion piiri eli a + b + c

Tutkimme kolmiosta seuraavia asioita:

1. Muodostavatko annetut sivut todella kolmion
2. Minkä tyyppisestä kolmiosta on kysymys
3. Mikä on sitten ala

Kohtaan 1 saamme vastauksen laittamalla ensin sivut suuruusjärjestykseen ja tarkastamalla sitten, onko kahden lyhimmän sivun summa pienempi kuin pisin sivu.

Kohtaan 2 saamme eri vastauksia seuraavasti:

Jos kaikki kolme sivua ovat yhtä pitkiä, on kyseessä tasasivuinen kolmio.
Jos kaksi sivua on yhtä pitkiä, on kyseessä tasakylkinen kolmio.
Jos kahden lyhyimmän sivun neliöiden summa on sama kuin pisimmän sivun neliö, on kyseessä suorakulmainen kolmio.

Algoritmi voisi olla seuraavanlainen:

Kolmion tutkiminen:

```
Annetaan kolmion sivut a, b ja c
Lajitellaan sivut suuruusjärjestykseen (lyhin ensin) taulukkoon
Sivut[3]
Lasketaan, onko Sivut[0] + Sivut[1] < Sivut[2]
Jos on, kyseessä on kolmio. Muutoin tulostetaan ('Ei kolmio'). Loppu
Verrataan, onko Sivut[0] = Sivut[1] = Sivut[2]
Jos on, tulostetaan ('Tasasivuinen')
Jos ei, verrataan, onko Sivut[0] = Sivut[1] OR Sivut[0] = Sivut[2]
OR Sivut[1] = Sivut[2].
Jos jokin ehto toteutuu, tulostetaan ('Tasakylkinen')
Jos ei, verrataan, onko Sivut[0]2 + Sivut[1] 2 = Sivut[2] 2 (sivut
olivat suuruusjärjestyksessä).
Jos kyllä, tulostetaan ('Suorakulmainen')
Lasketaan A = [s(s-a)(s-b)(s-c)]1/2.
```

9.2 Suora

Viiva tai vektori on kuva-alkio, joka sisältyy kaikkiin grafiikkasovelluksiin. Viiva-alkioiden avulla voidaan taas approksimoida mitä tahansa muita graafisia kuvioita. Tässä luvussa käsitellään hieman parametrisia menetelmiä *muodostaa* vektori ja ympyrä. Parametrista menetelmää kutsutaan myös nimellä *DDA* (Digital Differential Analyzer).

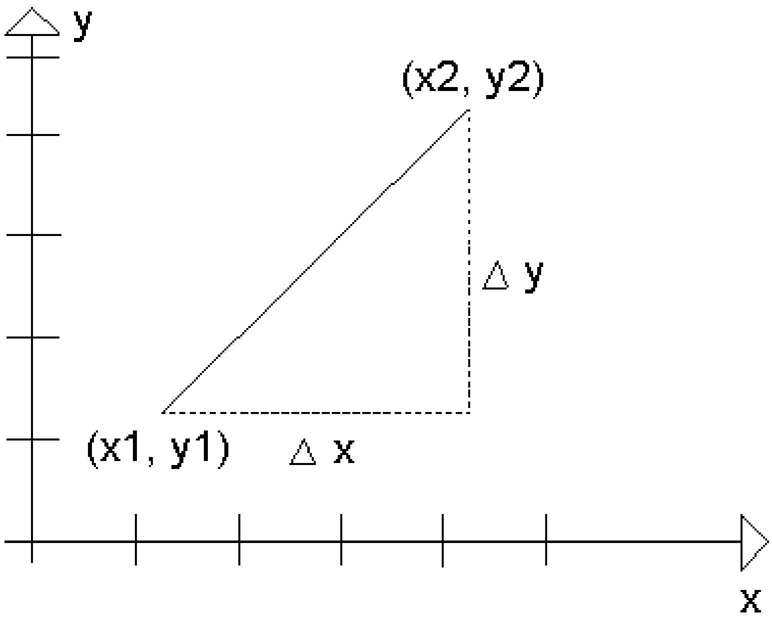
Kuten tiedetään, mikä tahansa tasokäyrä voidaan esittää parametrimuodossa.

$$x = x(t)$$
$$y = y(t)$$

(Parametrimuodon lisäksi käyrä voidaan esittää ratkaistussa muodossa $y = y(x)$ tai ratkaisemattomassa muodossa $f(x,y) = 0$).

Kahden pisteen (x1,y1) ja (x2,y2) välisen vektorin (katso kuva) parametriesitys on:

$$x = x1 + delta\_x\ t$$
$$y = y1 + delta\_y\ t$$

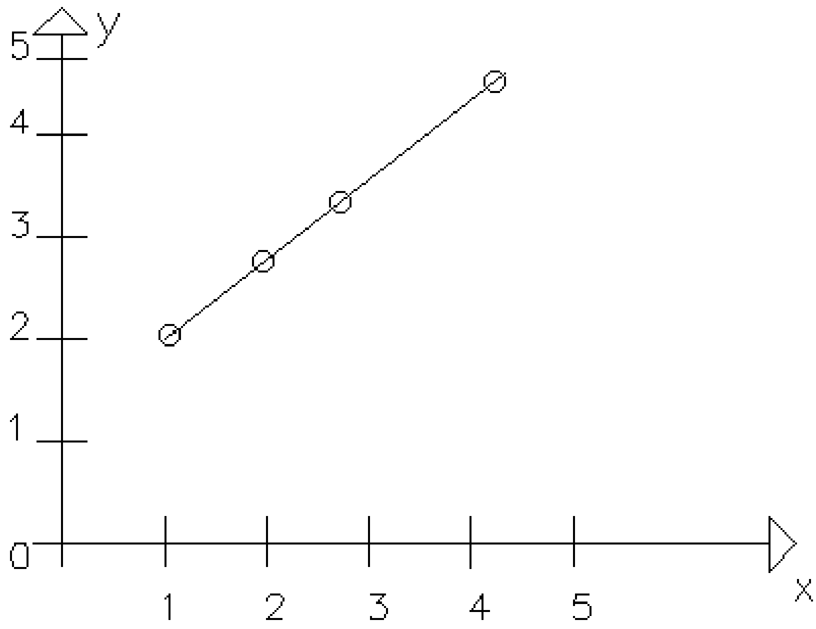


Vektorin parametrimuodon havainnollistamiseksi katsotaan seuraavaa esimerkkiä. Jos meillä on kaksi pistettä (1,2) ja (5,5), niin niiden välisen janan parametriesitys on:

$x = 1 + t\,(5-1) = 1 + 4t$
$y = 2 + t\,(5-2) = 2 + 3t$
$0 <= t <= 1$

Laskemme muutaman pisteen arvon ja piirrämme suoran koordinaatistoon.

t	x	y
0	1	2
1/4	2	2 3/4
1/2	3	3 1/2
1	5	5



(Jos poistamme yhtälöparista t:n sisältävät termit esimerkiksi kertomalla ylempi yhtälö 3:lla ja alempi -4:llä ja laskemalla yhtälöt yhteen saadaan 'normaali yhtälö', eli $y = 3/4\ x + 5/4$).

Parametrimuotoinen käyrä, $x = x(t)$, $y = y(t)$ voidaan esittää integraalia approksimoimalla muodossa

$$x = x1 + \Sigma dx \quad \text{ja}$$
$$y = y1 + \Sigma dy$$

Käyrän pisteitä generoidaan siis lisäämällä pieniä siirtymäosia (dx ja dy) lähtien annetusta alkupisteestä.

Vektorien kohdalla uusi piste saadaan edellisestä seuraavasti:

x2 = x1 + delta_x dt
y2 = y1 + delta_y dt

Yhtälöparissa dt kuvaa askelpituutta ja x- ja y-suuntaiset siirtymät ovat:

delta_x dt = dx
delta_y dt = dy

Koska koordinaattiarvot on pyöristettävä kokonaisluvuiksi, on dt valittava harkitusti, jotta vektoriin ei tulisi esimerkiksi katkoksia.

Jos dt:n arvoksi valitaan esimerkiksi 0.2, saataisiin edellä olevan janaesimerkin mukaan (pisteiden (1,2) ja (5,5) välinen jana) seuraavat askelarvot:

delta_x = 5-1 = 4
delta_y = 5-2 = 3

dx = delta_x dt = 4 * 0.2 = 0.8
dy = delta_y dt = 3 * 0.2 = 0.6

Siirtymät näyttävät melko sopivilta. dt:n arvo kannattaa kuitenkin sitoa esimerkiksi vektorin pituuteen, josta dt on tietty osa.

Seuraavana on algoritmiluonnos vektorin piirtämiselle.
(Funktio TulostaPiste(x,y,c) kuvaa pikselin piirtoa tiettyyn kohtaan.)

Viivan generointi:

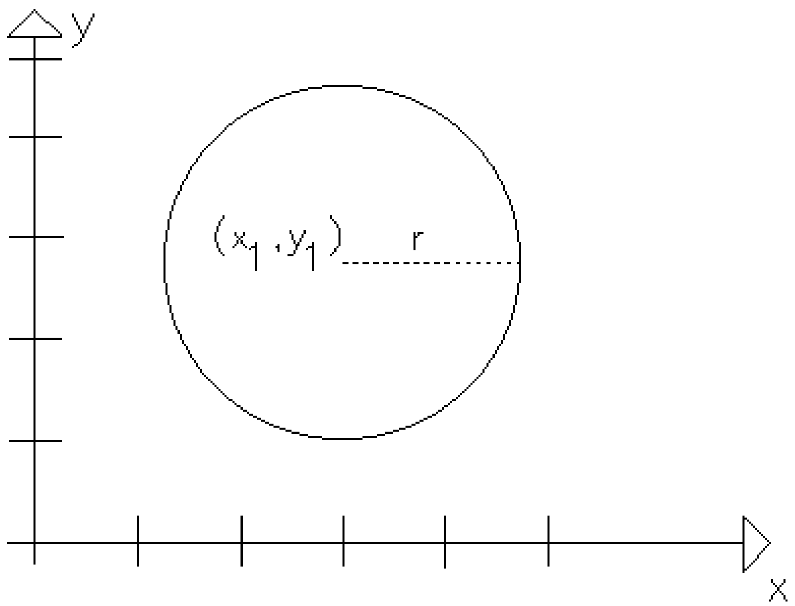
```
Olkoot alku- ja loppupisteet (x1, y1) ja (x2, y2)).
Esitellään muuttujat, joista delta_x ja delta_y ovat kokonaislukuja.
    delta_x = x2 - x1
    delta_y = y2 - y1
    dt = vektorin pituus/10
    dx = delta_x * dt
    dy = delta_y * dt
    x = x1
    y = y1
    TulostaPiste(x,y)
    i = 1 to vektorin pituus
        x = x + dx
        y = y + dy
        Pyöristä piste(x,y)
        TulostaPiste()
```

Algoritmia voi kehittää edelleen esimerkiksi siten, että x-siirtymän arvo määritetään kiinteästi 1:ksi ja sen perusteella lasketaan y-siirtymän arvo.

9.3 Ympyrä

Ympyrän parametrimuoto (olkoon keskipiste (x1,y1) ja säde r) on seuraava:

$x = x1 + r\cos t$
$y = y1 + r\sin t$
$0 <= t <= 2\pi$



Ympyrän yhtälö on siis

$r^2 = x^2 + y^2$

Jos ratkaisemme yhtälöstä y:n, saamme

$y =+- (r^2 - x^2)^{1/2}$

Derivointi antaa tuloksen

$y' = dy/dx = - x/(r^2 - x^2)^{1/2} = -2x/y$ (Koska $(r^2 - x^2)^{1/2} = y$)

Eli yhtälöstä suoraan saamme

$$dy = - dx * 2x/y$$

Jos nyt asetamme x-siirtymän dx kiinteästi 1:ksi, saamme käyttää y-siirtymän arvoa

$$dy = - 2x/y$$

Näillä tiedoilla saammekin ympyrän kehän pisteen edellisestä kehän pisteestä:

$$x2 = x1 + 1$$
$$y2 = y1 - 2x/y$$

Muuntamalla edellä olevaa vektorin algoritmia voimme piirtää myös ympyrän kehän samalla menettelyllä.

9.4 Muita käyriä

Käyttämällä napakoordinaatistoa (polaarisia koordinaatteja) voidaan erikoiskäyriä generoida suhteellisen helposti. Pisteen (x,y) paikka ilmaistaan napakoordinaatistossa seuraavasti:

$$x = r\cos\theta$$
$$y = r\sin\theta$$

Antamalla kulmalle θ askeleittain sopivat arvot (esimerkiksi väliltä $(-\pi...+\pi)$) ja määrittelemällä arvo r käyrän yhtälön mukaan, voidaan piirtää esimerkiksi spiraali.

Spiraalin yhtälö on:

$$r = \theta.$$

Algoritmissa on siis $r = \theta$ ja alkupisteeksi (x1,y1) voidaan valita vaikka näytön keskipiste. Pisteitä saadaan siis määriteltyä yhtälöillä:

$$x = x1 + r\cos\theta \text{ ja}$$
$$y = y1 + r\cos\theta.$$

Pisteiden väliset viivat voidaan piirtää tässä tapauksessa vaikka ohjelmointikielen grafiikkakirjaston funktioilla.

Ympyrä saadaan piirrettyä antamalla säteen arvo, ellipsin yhtälö on taas muotoa

$$r = 4/(2+\cos\theta).$$

Jotta käyrä saataisiin piirrettyä näyttöalueelle, voidaan käyrän alkupisteeksi määrittää esimerkiksi puolet vaaka- ja pystyresoluutiosta.

9.5 Geometriset muunnokset

Tärkeimpiä graafisiin kuvioihin kohdistuvia muunnosoperaatioita ovat skaalaus, kierto ja siirto. Käsittelemme näitä hieman seuraavaksi.

9.5.1 Siirto

Siirto-operaatiossa pisteelle (x,y) annetaan siirtoarvo koordinaattien x ja y lisäyksenä. Merkitään lisäystekijöitä Lx:llä ja Ly:llä.

Uusi piste on siis:

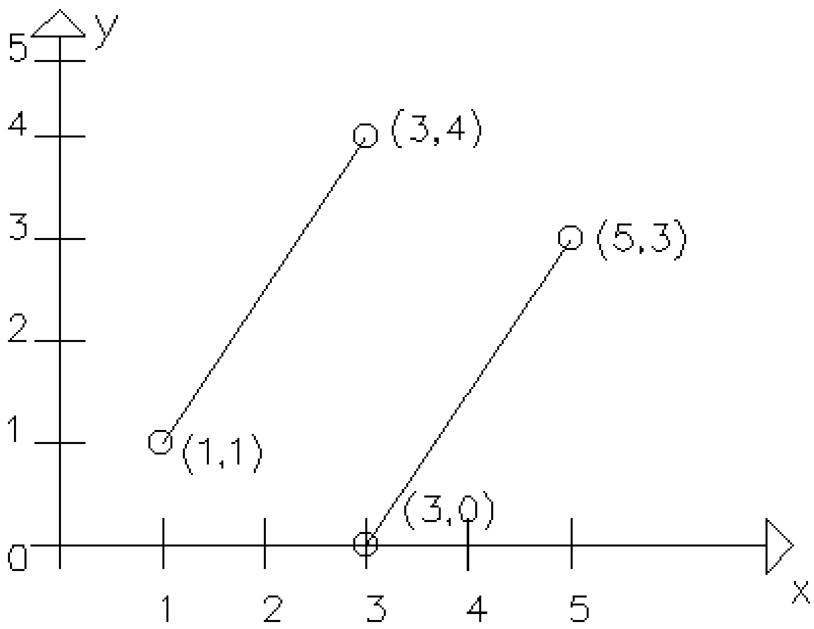
$$x2 = x1 + Lx$$
$$y2 = y1 + Ly$$

Seuraava esimerkki havainnollistaa tapahtumaa:

Olkoon kohteena jana, jonka alku- ja loppupisteet ovat (1,1) ja (3,4).
Siirtoarvot olkoot

$$Sx = +2 \text{ ja}$$
$$Sy = -1.$$

Kuten kuvasta näemme janan alku- ja loppupisteiksi tulevat (3, 0) ja (5, 3).



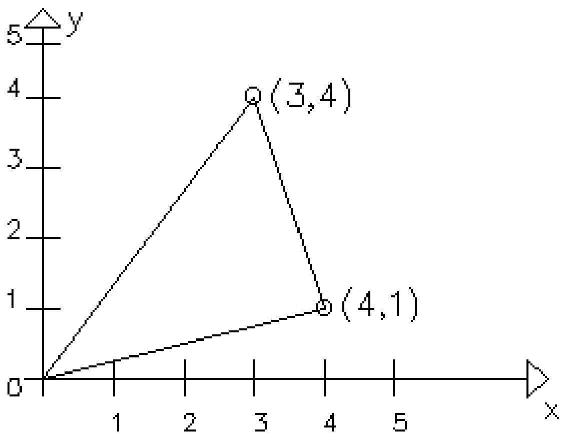
Samalla nähdään, että suoran siirtämiseen riittää, kun suoran päätepisteet siirretään.

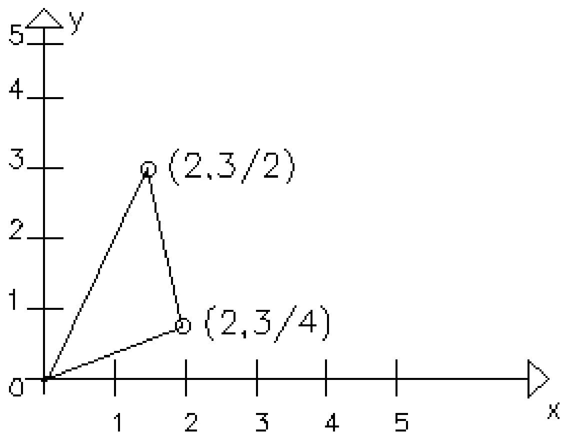
9.5.2 Skaalaus

Skaalauksessa pisteiden koordinaatit kerrotaan skaalaustekijöillä.
Tällöin pisteiden koordinaatit muuttuvat seuraavasti:
Olkoon skaalaustekijät Sx ja Sy.

x2 = x1 * Sx
y2 = y1 * Sy

Kuva havainnollistaa skaalausta:
Esimerkissämme Sx = 1/2 ja Sy = 3/4





9.5.3 Kierto

Pisteen *kiertoa* käsitellään seuraavaksi vain origon suhteen. Pisteen paikka origon suhteen saadaan lausekkeista:

$$x = r \cos\alpha$$
$$y = r \sin\alpha$$

Lausekkeissa α kuvaa kulmaa x-akselista vastapäivään. Jos pistettä kierretään origon ympäri (edelleen vastapäivään) kulman β verran lähtien alkusijainnista, saadaan uuden pisteen paikka x- ja y-koordinaatteina lausekkeista:

$$x2 = x1 \cos\beta - y1 \sin\beta$$
$$y2 = x1 \sin\beta + y1 \cos\beta$$

Antamalla kulmalle β pieniä siirtymäarvoja saataisiin ylläolevilla kaavoilla piirrettyä myös origokeskeisiä ympyröitä.

Ohjelma-algoritmit on helppo kehittää aiemmin laaditun vektorin piirtoalgoritmin pohjalta.

Tehtäessä operaatioita tasossa satunnaisen pisteen kautta, tehdään yleensä aina ensin siirto origoon, jonka jälkeen suoritetaan muunnosoperaatiot. Lopuksi kuvio siirretään takaisin alkuperäiseen paikkaan muunnettuna. Tällaista proseduuria helpottaa, kun käytetään matriisimuotoja ja matriisimatematiikkaa.

9.6 Rekursiiviset kuviot

Rekursiiviset kuviot muodostuvat ns. 'fraktaaligeometrian' mukaisesti eli niitä muodostettaessa ei käytetä tarkoin määriteltyjä matemaattisia (usein monimutkaisia ja hankalia) yhtälöitä, vaan lähes satunnaisuutta. Rekursiivisuuden mukaisesti muodostuvasta kuvion osasta tulee saman muodostamisfunktion sisäänmeno. Rekursiiviset kuviot ovat herättäneet viime aikoina huomiota. Niiden avulla on voitu hämmästyttävällä tavalla muodostaa luonnossa esiintyviä kuvioita kuten pilvet ja puut, mutta myös mitä tahansa muita kuvioita.
Tällaisten kuvioiden perusta on itse asiassa muunnoksissa, joista osa esiteltiin edellä. Jos lähtökuviona on esimerkiksi piste, voidaan rekursiivisia kuvioita saada aikaan tekemällä pisteelle lukuisia muunnoksia ja tulostamalla pisteet näytölle. Kuuluisa rekursiivinen kuvio, Sierpinskin kolmio, käyttää itse asiassa vain kolmea muunnosta. Muunnosten suorittamistaajuuteen ja -järjestykseen vaikutetaan antamalla muunnoksille todennäköisyysarvo (0...1), jolloin esimerkiksi todennäköisyysarvolla 0.1 tapahtuva muunnos suoritetaan harvemmin kuin 0.4 arvolla oleva muunnos.

Kuvion pisteitä voidaan muodostaa yhtälöillä:

$$X\_uusi = a X + b Y + c$$
$$Y\_uusi = d X + c Y + f$$

Kirjaimet a, b, d ja e kuvaavat nimenomaan muunnosten arvoja. Koska kyseessä on kertolasku, ne kuvaavat skaalausta ja kiertoa. Tekijät c ja f kuvaavat siis siirtoa.

Lukija voi kokeilla muodostaa rekursiivisia kuvioita antamalla esimerkiksi alkupisteen tai vaikkapa janan, jonka uudet pisteet saadaan em. yhtälöillä. Tekijöille a ... f voi koettaa antaa eri arvoja. Tulostus on suoritettava tarpeeksi monelle pisteelle (ehkä sadoille). Pisteen värin voi sitoa esimerkiksi suoritettavaan muunnokseen. Satunnaislukugeneraattori voi valita tehtävän muunnoksen, mihin voidaan liittää muunnoksille määritetyt todennäköisyydet.

9.7 Bezier-käyrä

Otamme grafiikkaluvun lopuksi esille erään kehittyneen käyrätyypin, nimittäin Bezier-käyrän. Bezier-käyrä on myöskin parametrimuotoinen käyrä, jossa parametri t saa arvoja väliltä 0...1. Käyrä kehitettiin jo 1970-luvulla lähinnä autoteollisuuden tarpeisiin ranskalaisen matemaatikon, Bezierin, toimesta. Käyrän avulla haluttiin määrittää mm. autojen pintamuotoja.

Bezieriä kuvaava yhtälö on polynomiyhtälö, jonka asteluku voi olla mielivaltaisenkin suuri. Asteluvun valintaa rajoittaa tietenkin tietokoneen suoritusteho.

Kolmannen asteen (kuutiollisen) Bezier-käyrä voidaan esittää yhtälöillä:

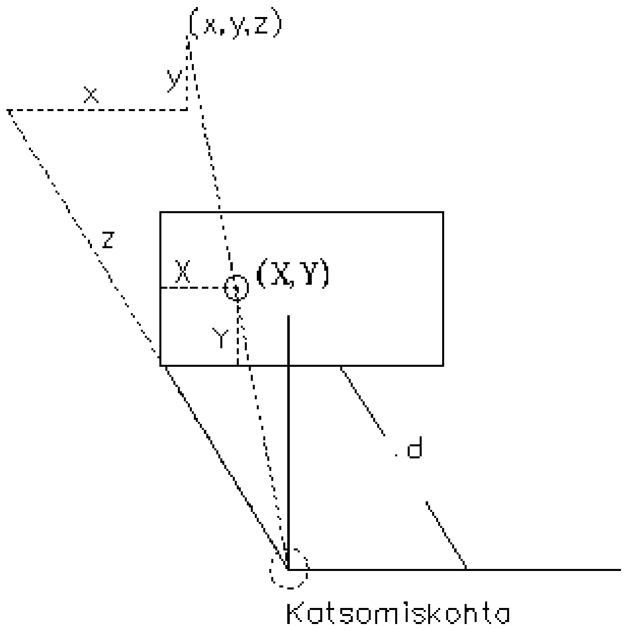
$$x(t) = (1-t)^3 x_1 + 3t(1-t)^2 x_2 + 3t^2(1-t)x_3 + t^3 x_4 \text{ ja}$$
$$y(t) = (1-t)^3 y_1 + 3t(1-t)^2 y_2 + 3t^2(1-t)y_3 + t^3 y_4$$

Piste (x_1, y_1) kuvaa käyrän alkupistettä ja (x_4, y_4) taas loppupistettä. Pisteet (x_2, y_2) ja (x_3, y_3) ovat niin sanottuja ohjauspisteitä, jotka sijaitsevat kehyslaatikon vasemmassa ja oikeassa yläkulmassa. Käyrän alku- ja loppupisteitä voidaan samalla pitää kehyslaatikon alimpina ohjauspisteinä. Bezier-käyrä määrittyy siis eräänlaisen laatikon sisään. Käyrä ei mene missään laatikon ulkopuolelle.
Käyrän hyviä puolia on mm. se, että käyrään voidaan kohdistaa geometrisia muunnosoperaatioita, joita käsittelimme aiemmin tässä luvussa.

Käyrän pisteitä saadaan antamalla t:lle arvoja väliltä 0...1 ja laskemalla vastaavat x ja y.

9.8 Hieman 3D-kuvioista

3D-kuvioita joudutaan yleisesti esittämään 2D-pinnalla. Jotta kuvio hahmottuisi oikein, käytetään apuna perspektiivejä. Tällöin haetaan piste, jota kohti kuvio kapenee, jolloin kuvio koetaan 3-ulotteiseksi. Suoraviivaisten kuvioiden generoiminen näin ei ole vielä ongelma, mutta kuvioiden tullessa monimutkaisemmiksi niiden määrittäminen tulee vaikeammaksi. Myös katsomissuunnan valinnalla on merkitystä kuvion määrittämisen vaikeusasteeseen. Kuvamme havainnollistaa perspektiiviajatusta. Kuvassa kuvatasossa olevaa pistettä A (X,Y) katsotaan origosta päin ja sitä vastaa kauempana oleva piste P (x,y,z).



Pisteiden A ja P koordinaattien suhde on siis:

X = x*d/z
Y = y*d/z

Tällä menettelyllä voidaan muodostaa suoraviivaisia 3D-kuvioita määrittelemällä ensin kauempana olevat pisteet ja projisoimalla ne lähemmäksi (kuvatasolle). Lopuksi pisteet yhdistetään viivalla.

Yleisesti mikä tahansa pinta voidaan esittää ratkaistussa muodossa, z = z(x,y), ratkaisemattomassa muodossa, f(x,y,z) = 0 tai parametrimuodossa, x = x(t,u), y = y(t,u), z = z(t,u). Avaruuspintaa esittää yksi yhtälö. Avaruuskäyrä saadaan esitettyä kahden pinnan leikkauskäyränä eli tällöin riittää kaksi yhtälöä kuvaamaan avaruuspintaa. Esimerkiksi 'kierukka' saadaan esitettyä yhtälöillä x = cosz, y = sinz. Yhtälö $z = x^2 - y^2$ määrittelee erään satulapinnan. Ratkaisemalla yhtälöstä x ja y voidaan pinta piirtää kuvaruudulle. Kaikissa generoinneissa on päätettävä katselusuunta sekä projisioitava pisteet kuvatasolle ylempänä esitettyjä koordinaattien suhteita käyttämällä.