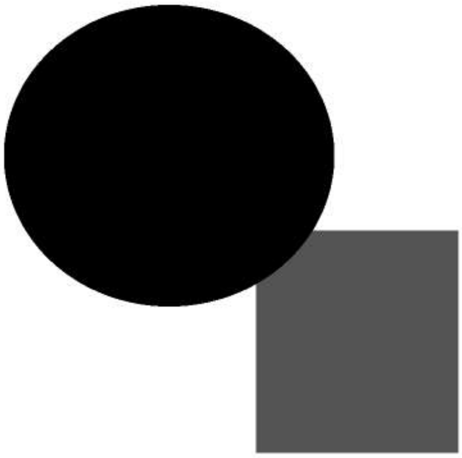
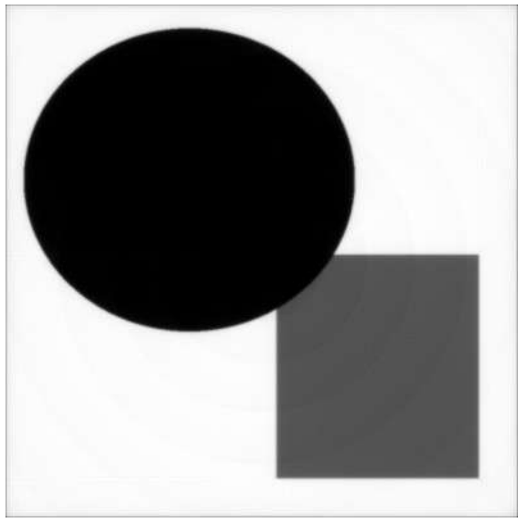
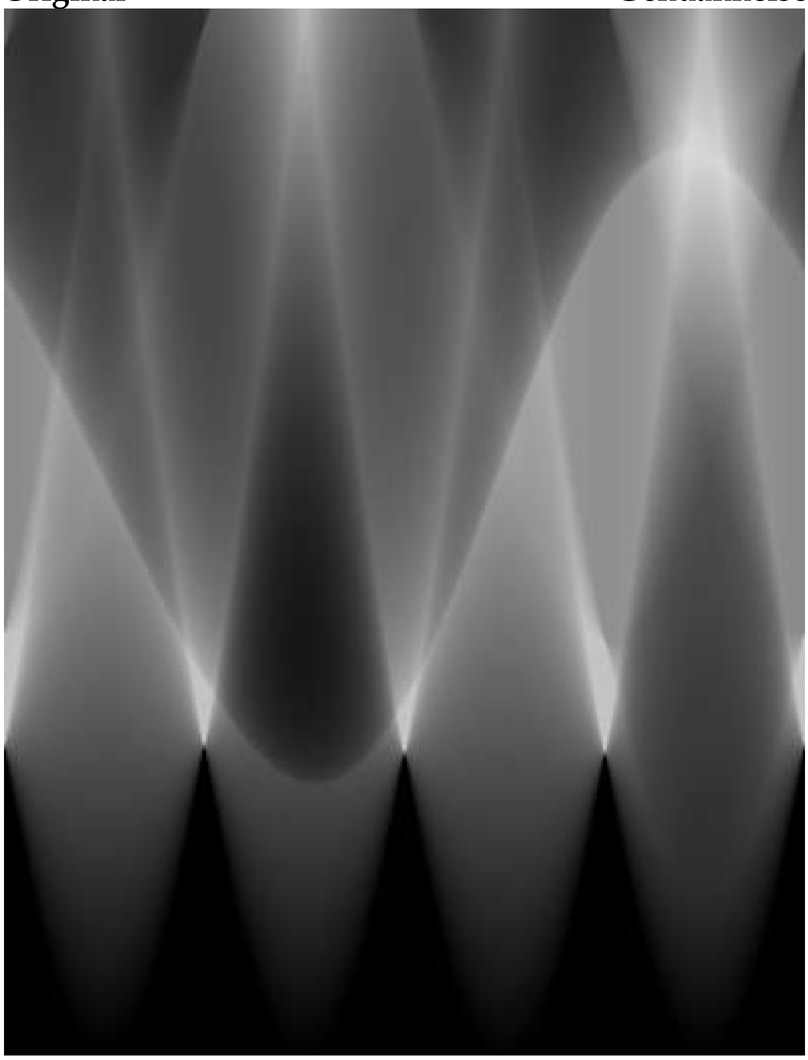


Original



Gendannelse



Grafisk gengivelse af skanningen

Tomografi (röntgen-skanner-princippet)

I 1970-erne var tiden moden til at skabe et apparatur som man længe havde været sikker på var en teknisk mulighed: et röntgenapparat som gør det muligt at se *rigtigt* ind i legemer som er uigennemtrængelige for lys - i modsætning til det ældgamle röntgenbillede hvor det kun er en *projektion* af det indre man kan se. Problemet er af to-dimensional natur, siden man kun behøver at danne billedet af et plant snit igennem legemet. Og dette må nødvendigvis ske ved at sende röntgen- eller lignende stråler igennem legemet fra et stort antal retninger beliggende i den pågældende plan, og for hver retning måle hvor meget strålens intensitet aftager, og så ud fra alle disse informationer konstruere et billede. Men hvordan? Ved det sædvanlige röntgenapparat udsendes stråler fra röntgenrøret i alle retninger indenfor en kegle, og disse kan opfanges på en fotografisk plade eller på en glasplade som flourescerer der hvor den rammes af en stråle. Men nu skal der tilsyneladende også matematik til for at danne billedet: informationerne skal omdannes til tal som skal bearbejdes i formler, og ud fra de bearbejdede tal skal man få et billede frem på en skærm. Dette sidste var ikke noget problem, det havde man tv-skærme til, men det første problem kunne muligvis være formidabelt: der skal øjensynligt benyttes informationer fra "alle tænkelige" linier indenfor det plane snit, og da en linie er karakteriseret ved en vinkel og en forskydning, er der måske i praksis brug for 300x300 = 90.000 linier, og for enhver af disse fås et tal (intensitetstabet) som skal opbevares da det skal bruges for alle punkterne på skærmen. Og denne skærm er måske på 400x400 = 160.000 punkter. Altså: 90.000 tal skal oplagres i en hukommelse og bearbejdes for hvert af de 160.000 punkter, og til enhver sådan bearbejdning behøves måske flere tusinde operationer. For at danne et fraktalbillede skal der ikke oplagres nogen tal, og for hvert punkt skal der oftest kun udføres nogle hunderede udregninger.

Og dog var det ikke dette oplagrings- og regneproblem man skulle slås mest med i 70-erne, har jeg ladet mig fortælle. Man rådede allerede dengang over computerkraft som var stor nok. Nej, det var *matematikken:* man kendte ikke de formler der skulle bruges. Det var især opstillingen af disse der var den store bedrift af de to medicinere som i 1979 fik Nobel-prisen for at have løst problemerne. Kan dette virkelig være sandt? Matematikerne har alle dage hævdet, at de fra deres elfenbenstårn har et glimrende udsyn over verden og at de er langt forud for denne i udvikling. Og når nogen har sat spørgsmål ved nytten af deres arbejde, har de altid forsvaret dette med at det *engang* vil vise sig nyttigt. Og her var der endda tale om et problem hvis løsning måtte være langt under deres niveau og som de længe burde have vidst at verden snart ville få brug for.

Og selvfølgelig var et så elementært problem som der her er tale om, løst forlængst. Det er et centralt problem indenfor det fagområde der hedder *integralgeometri,* og denne disciplin havde eksisteret i over hundrede år. Og i 1917 offentliggjorde den franske matematiker J.K.A. Radon (1887-1956) en afhandling, hvori han fremlagde og beviste de formler der skal til for at rekonstruere en funktion defineret på et område i planen ud fra dens *integral* langs alle tænkelige linier igennem området. Og hvis denne funktion er tætheden (i en eller anden forstand) af et materiale fordelt over det plane område, så må integralet langs en ret linie være et mål for den *modstand* som en stråle der er følsom overfor materialet møder, når den trænger igennem materialet langs linien, dvs. den *aftagen* i intensitet der sker i en strålen.

Radon har dog næppe tænkt den tanke at hans teori i princippet kan bruges til at se ind i legemer. Röntgenbilleder havde man ganske vist haft i tyve år, men fra denne matematikløse teknik til hans regnekrævende formler er der lang vej. Radon havde udtænkt sin formel fordi den på hans tid var en oplagt udfordring, et naturligt trin i integralgeometriens udvikling, og måske viste formlen og dens bevis sig ligefrem at besidde en stor skønhed. Om formlen vil jeg sige, at den ser ud som enhver erfaren matematiker vil forvente, den er enkel og ganske køn. Det er mere beviset som den matematikinteresseret hungrer efter at se, og her vil han blive glædelig overrasket. Men inden vi skitserer teorien skal vi formulere Radons formel og efterprøve den på computeren: hvor lang tid tager det mon at danne billedet? - hvordan bliver det?

Radons formel lyder således (når den formuleres ud fra *vores* problemstilling):

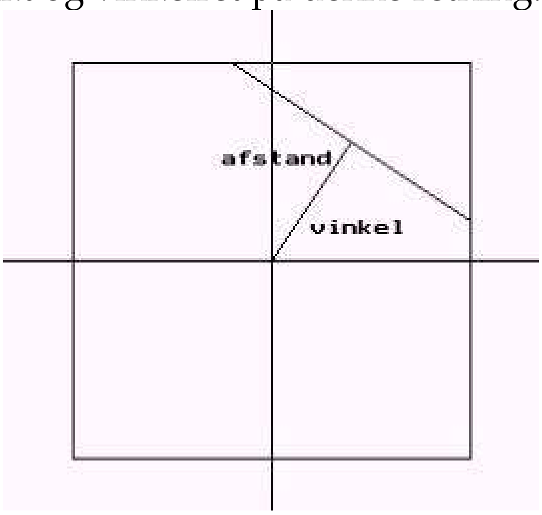
> For ethvert punkt p i området defineres en funktion M(p, r) af de positive reelle tal r ved:
>
> M(p, r) =
> middelværdien af modstanden i alle de rette linier som har afstand r fra p.
>
> Da gælder:
>
> $$\text{materialets tæthed i p} = -1/\pi \int_0^D (dM(p, r)/dr)\, dr/r,$$
>
> hvor D er områdets diameter.

Vi vil forestille os at vi indenfor et rektangulært område har en fordeling af masse, men dette svarer til et billede med gråtoner, idet hver gråtone tillægges en tæthed. Da computerens farver er bestemt ved tripler (R, G, B) af hele tal fra 0 til 255 (nemlig farvens sammensætning af rød, grøn og blå), og da en gråtone er en sådan farve hvor de tre tal er ens, kan vi lade tætheden være dette tal. Tætheden er altså et helt tal fra 0 til 255. Billedet kan f.eks. være et fotografi. Vi laver to programmer: det ene ("Skan") "skanner" billedet og lagre disse informationer i en fil (af reelle tal), det andet ("Dan") gendanner billedet ud fra informationerne i filen, ved for hvert af punkterne at udregne den ovennævnte formel og tildele punktet den gråtone som tallet angiver. Skanningen foregår ved at vi for et stort antal rette linier igennem rektanglet tager summen af tæthederne for de pixel linien går igennem.

Vi må gøre os klart at vi må acceptere nogle fejl i det billede vi genskaber, da vi jo har med integrationer og differentationer at gøre og disse kun kan udføres med tilnærmelse. Og skal vi have en stor præcision vil programmet nok køre urimeligt langsomt. At der kan blive fejl kan vi tænke os til på forhånd: på de steder hvor der i den oprindelige figur er bratte overgange, vil der i det rekonstruerede billede kunne være nogle fejlfarvede punkter, og hvis den oprindelige figur er kompliceret, f.eks. sammensat af mange ens figurer ligesom et ark frimærker, vil der ved skanningen blive en tendens til at modstanden er næsten ens i alle retninger, så det rekonstruerede billede bliver grumset. På den anden side, da Radons formel jo er helt eksakt og da en moderne computer er hurtig, kan vi godt tillade os at stille store krav til billedet. At det måske vil tage en god tid at genskabe billedet, skal vi ikke bekymre os om, dette har ingen betydning hvis nogen vil bruge vores computerprogram i praksis. Ved skanningen vil man ikke bevæge én stråle i tusinder af stillinger, men vil sende tusinder af stråler afsted samtidig. Og ved dannelsen af billedet vil man ikke udføre operationerne *lineært* således som vi gør i vores program, men *parallelt,* så udregningerne udføres samtidig for flere punkter, og så billedet genskabes på en brøkdel af et sekund.

Vi vil først udtænke et program som er simplest muligt, idet vi udfører integrationerne og differentationerne lige efter lærebogen, dvs. tager sådanne tilnærmelser som optræder i deres definition. Så vil vi undervejs opdage, at vi med enkle midler kan forbedre programmet betydeligt.

Skanningen kan udføres på flere måder, og det har ikke stor betydning hvilken man anvender. I mit program har jeg defineret en funktion *modstand(vinkel, afstand)* på følgende måde: for en given *vinkel* og *afstand* går man stykket *afstand* ud fra rektanglets centrum i retningen *vinkel* og betragter den rette linie igennem dette punkt og vinkelret på denne retning:



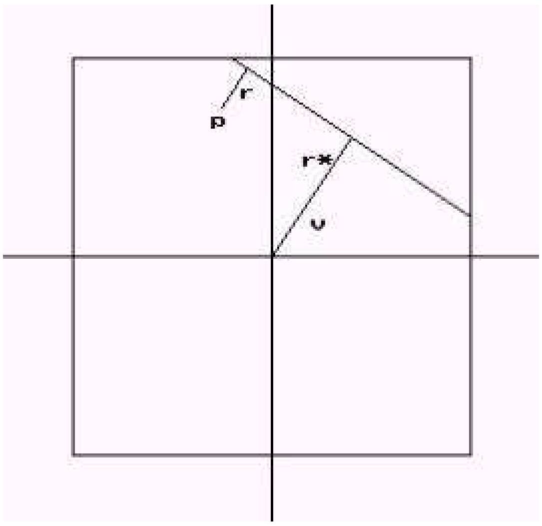
Den modstand som denne stråle møder ved at passere igennem figuren, fås ved at addere tallene mat[i, j] for alle de punkter (i, j) den rammer. For at kunne finde disse punkter må vi lade den ene af koordinaterne i og j være uafhængig variabel og udregne den anden ud fra liniens ligning. Vi vælger i (x-koordinaten) for de linier hvor *vinkel* er nærmest y-aksen (dvs. $|\cos(\mathit{vinkel})| < |\sin(\mathit{vinkel})|$), og j (y-koordinaten) for de linier hvor *vinkel* er nærmest x-aksen (dvs. $|\cos(\mathit{vinkel})| > |\sin(\mathit{vinkel})|$). Men da modstanden i linien er et integral langs linien, skal vi gange det tal vi har fundet med længden af en pixel, denne længde er konstant langs linien, men den er ikke den samme for de forskellige retninger: en pixel er jo et lille kvadrat, og når linien ligger skå i forhold til dette, må den *i gennemsnit* passere en længere vej. Derfor skal vi i det første tilfælde dividere med $|\sin(\mathit{vinkel})|$, og i det andet tilfælde dividere med $|\cos(\mathit{vinkel})|$.

Ved hjælp af denne funktion *modstand(vinkel, afstand)* kan vi nu skanne figuren, dvs. finde modstanden i et stort antal *udvalgte* linier, f.eks. lade *vinkel* gå fra 0 til 359 og lade *afstand* gå fra 0 til rektanglets radius R igennem f.eks. 300 trin - så har vi fået "alle tænkelige" positioner med.

Herefter skal vi for hver pixel først finde funktionen M(p, r) og derefter udregne integralet

$$\int_0^{2R} (dM(p, r)/dr)\, dr/r.$$

For et givet r > 0 er M(p, r) summen af modstanden i alle de udvalgte linier som har afstanden r fra punktet p divideret med antallet af disse linier (altså 360 - selv om en linie ligger helt udenfor rektanglet skal den tælles med alligevel). For enhver udvalgt vinkel v skal vi finde den udvalgte linie hvis afstand fra p er nærmest r. Det gøres på følgende måde: Hvis punktet p har koordinatsættet (a, b) er den rette linie som har afstanden r fra p og som er vinkelret på retningen v, den samme som den rette linie som har afstanden $r^* = r + a\cdot\cos(v) + b\cdot\sin(v)$ fra origo og som er vinkelret på retningen v (idet $a\cdot\cos(v) + b\cdot\sin(v)$ er projektionen af linien fra origo til p ind på retningen v):



- med mindre at r* bliver negativ, for så skal vi lægge 180 til v, da vi har valgt at operere med vinkler hele vejen rundt, men kun med positive afstande. Derfor er den søgte udvalgte linie, linien givet ved den udvalgte vinkel v og den udvalgte afstand som er nærmest tallet r*, eller, hvis dette tal er negativt, linien givet ved den udvalgte vinkel v+180 og den udvalgte afstand som er nærmest tallet -r*.

Herefter skal vi udregne det viste integral. I integranden ophæver de to dr-er hinanden, så vi skal blot dele intervallet fra 0 til 2R op i et stort antal lige store stykker - f.eks. 600 - og hvis vi kalder længden af hvert stykke d, skal vi for stykket der går fra r til r+d udregne tallet

$$(M(p, r+d) - M(p, r))/(r+d/2),$$

og alle disse 600 tal skal adderes. Hvis summen kaldes s, skal vi i programmet først sætte s = 0 og r = 0, og succesivt gøre følgende indtil r > 2R: udregne (M(p, r+d) - M(p, r))/(r+d/2), addere dette tal til s, erstatte r med r+d. Til sidst skal vi gange s med $-1/\pi$, afrunde det tal vi får til et helt tal og tildele pixlen farven svarende til dette tal.

Da dette primitive program kan forbedres betydeligt ved beskedne ændringer, vil vi straks gå igang hermed.

Skanningen kan der ikke gøres meget ved. Ikke udover at øge nøjagtigheden ved at gøre de små bidder som en linie deles i kortere, f.eks. halv så lange. Dette svarer til at lade figuren være dobbelt så detaljeret, og dette svarer igen til at forestille sig at den er dobbelt så mange pixel på hver led. Men kun den figur vi skanner, billedet vi danner skal stadigvæk kun være det oprindelige antal pixel på hver led.

Herefter kommer vi til differentationen og integrationen. Den inddeling vi foretager af intervallet [0, 2R] bør klart være finere jo nærmere vi kommer til 0, da integranden må formodes at være størst her (bemærk at der er en division med r). Vi vil begynde med et meget lille d, og succesivt gøre d større ved at erstatte d med kd, hvor k er et fast tal lidt større end 1 (f.eks. k = 1.1). Men vi bør også undlade at gøre brug af vores opdagelse af at dr optræder i både tæller og nævner af integranden (dM(p, r)/dr) dr/r. Vi bør operere med to forskellige dr-er. Dannelsen af en differenskvotient der er en tilnærmelse til dM(p, r)/dr i punktet r, bør foretages ud fra r-værdier beliggende på hver side af r: dM(p, r)/dr ≈ (M(p, r_2) - M(p, r_1)/(r_2 - r_1), hvor $r_1 < r < r_2$. Og i stedet for bare at lade bidraget til summationen fra r til r+d være d gange (dM (p, r')/dr')/r' taget i et punkt r' imellem r og r+d, vil benytte tilnærmelsesformlen:

$$\int_r^{r+d} f(r)\, dr/r \approx (f(r)/(r+d/4) + f(r+d)/(r+3d/4))\, d/2$$

som bygger på værdierne af f(r) i de to endepunkter af det lille interval [r, r+d] (den er ikke svær at eftervise). Så bliver denne inddeling af intervallet [0, 2R] forskudt i forhold til den inddeling vi bruger til at danne differenskvotienterne (i programmet betegner r1 de succesive r-er brugt til at danne differenskvotienterne, idet r1 begynder med r1 = 0, og r betegner de succesive r-er brugt til at danne summationen, idet r begynder med r = d/2). De to første trin af denne succesive procedure er lidt anderledes end de følgende, derfor optræder et z som først er 0, derefter 1, og resten af vejen 2. En ny differenskvotient kaldes hele tiden g samtidig med at den foregående får navneforandring til h. For z = 0 dannes den første, for z = 1 dannes den anden og for z = 2 dannes alle de følgende.

Der er en finesse mere i vores program: for at danne M(p, r) skal vi for enhver udvalgt vinkel v bruge modstanden i *den* udvalgte linie som har en afstand fra p der er *nærmest* ved tallet r. Her bør vi selvfølgelig indføre en interpolation: hvis r^*_1 og r^*_2 er de udvalgte afstande på hver side af den ovenfor fundne afstand r* fra 0, bør vi lade modstand(v, r*) være

modstand(v, r^*_1) + (r* - r^*_1)(modstand(v, r^*_2) - modstand(v, r^*_1))/(r^*_2 - r^*_1).

Og til allesidst en lille tidsbesparende ting: i integrationen behøver vi ikke at integrere helt til tallet 2R, som er rektanglets diameter, vi kan nøjes med at integrere til R+afstanden fra 0 til p.

Programmets virkemåde

I vores program vil vi i stedet for et billede i sort/hvid anvende et billede i farver - dette vil ikke gøre programmet mere kompliceret. Et billede i farver svarer nemlig til tre billeder i sort/hvid: intensiteten af rød, grøn og blå, og skanningen og gendannelsen af disse tre billeder kan foretages i én og samme procedure. For hver pixel i gendannings-proceduren giver Radons formel os tre hele tal R, G, B (fra 0 til 255), og punktet tildeles farven med disse RGB-værdier.

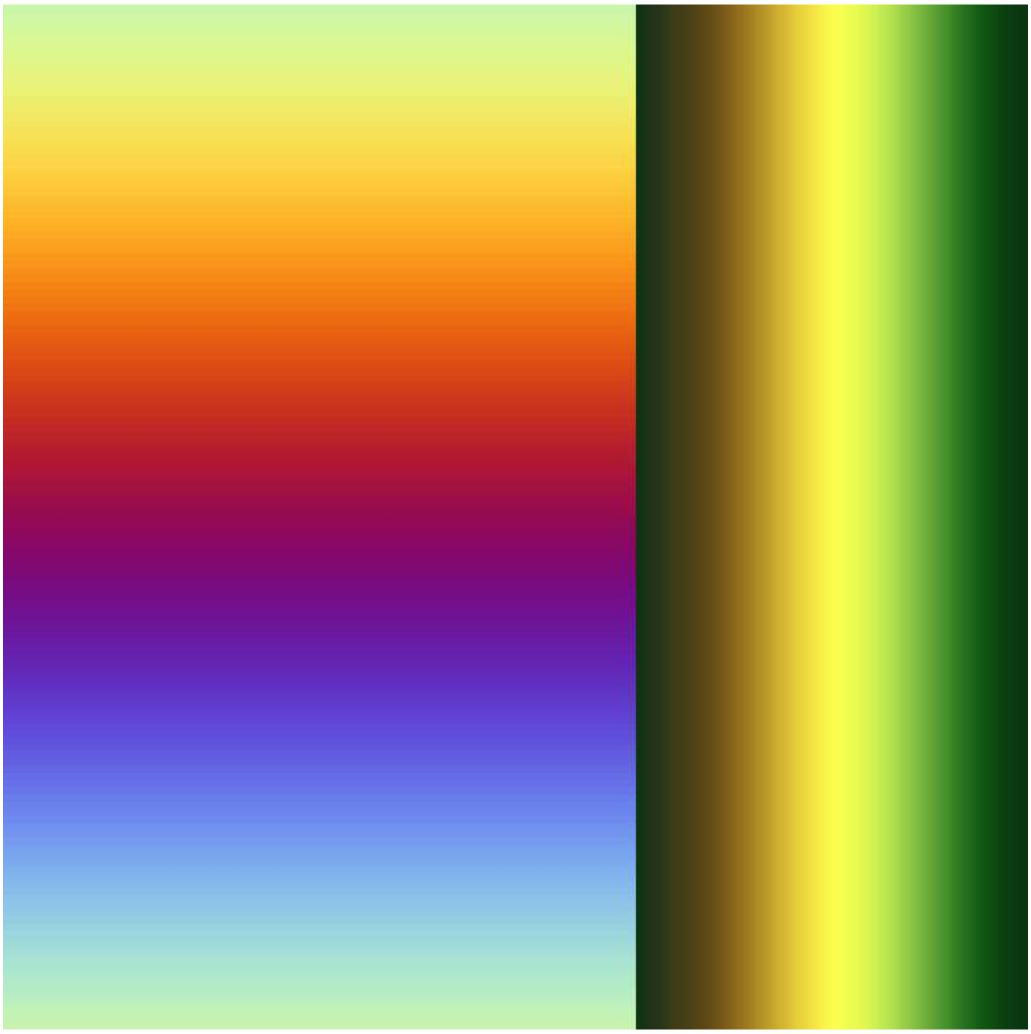
Vi forudsætter at billedet er i BMP-format og ikke for stort (f.eks. 600x600 pixel), og det gives navnet "Billede.bmp". I programmet "Skan" skal inden skanningen begynder indtastes antallet af vinkler og forskydninger (f.eks. henh. 400 og 300). Resultatet af skanningen er en fil af reelle tal, og den har navnet "Fil.julia". Programmet "Dan" tegner nu, ud fra informationerne i denne fil, et billede i halv så stor størrelse. Hvor fejlfrit det bliver, afhænger af antallet af vinkler og forskydninger, men også de af de tilnærmelser der er i programmet, og det vil altid blive lidt sløret:

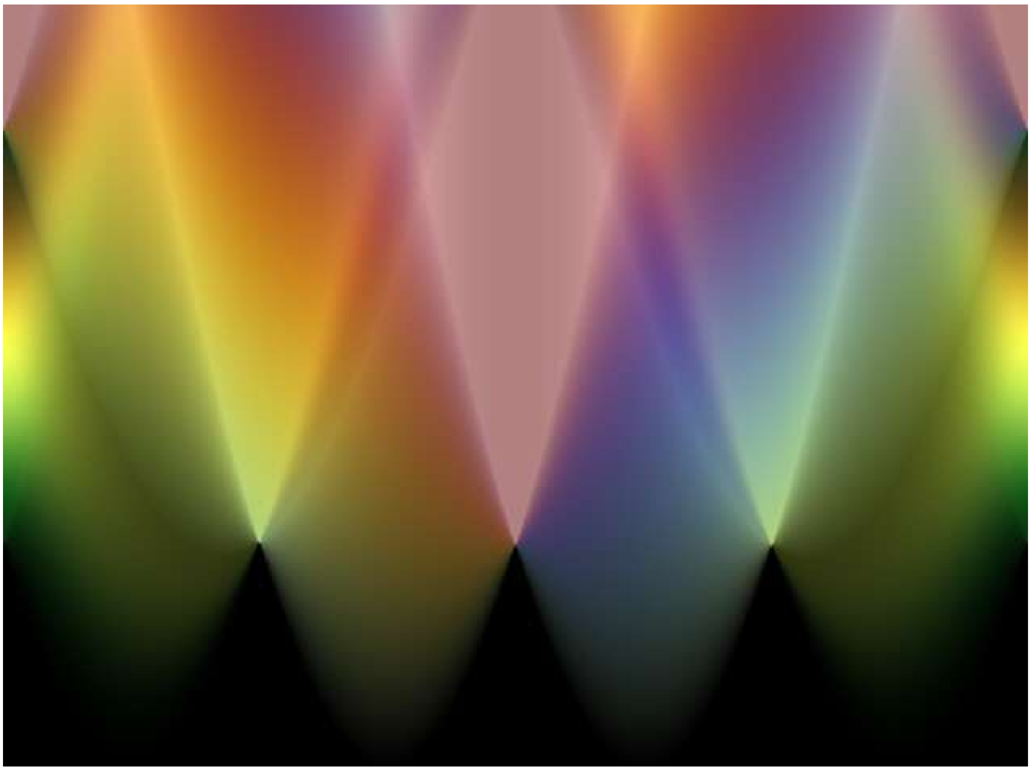

***

Programparret "Skan2" og "Dan2" er en variant af dette program, idet resultatet af skanningen i stedet for at være en fil af tal, er et billede (i BMP-format): tallene er afrundede til hele tal og justeret således at det største er 255, og vinklen og forskydningen er afsat ud af henh. den vandrette og den lodrette akse. "Dan2" danner billedet ud fra farveværdierne i dette billede (som er givet navnet "TransBillede" – billedet til venstre nedenfor). Ved, forinden at vi anvender "Dan2", at ændre lidt i dets farver (med et billedbehandlingsprogram), får vi andre farver i resultatet:


Da skanningen af et billede ikke "véd" hvor stort det oprindelige billede er, må dettes dimensioner (bredde og højde) indtastes ved gendannelsen (i programmet "Dan2"). Indtaster man andre dimensioner (hvor størrelsesforholdet er korrekt) bliver billedet lysere eller mørkere, derfor må også indtastes en "farvefaktor" – er billedet f.eks. halvt så stort skal den sættes til 0.5.

Ved at tage udgangspunkt i et simpelt billede, kan vi ved skanning få et "kunstværk" frem:





Bevis for Radons transformationsformel

Jeg vil nu skitsere Radons bevis for sin formel. Denne formel kaldes også for *Radon-transformationen,* og den hører blandt en type af matematiske formler som er særligt fascinerende, nemlig inversionsformlerne (i afsnittet om Riemanns primtalsteori omtales Möbius- og Fourier-inversionen).

Inversionsprincippet kommer ind i billedet i forbindelse med en vis *integralligning,* som vi får brug for at løse, dvs. en ligning hvor den ubekendte er en funktion som indgår i en integraldannelse. Og den integralligning som skal løses lyder: givet en funktion g(y), find en funktion f(x) således at

$$g(y) = \int_y^\infty f(x)/\sqrt{(x-y)}\, dx.$$

Men denne integralligning blev løst af *Niels-Henrik Abel* (1802-29), og løsningsformlen er ganske tankevækkende:

$$f(x) = -1/\pi \int_x^\infty g'(y)/\sqrt{(y-x)}\, dy$$

- vi skal altså blot differentiere g(y) og bruge den samme formel (og dividere med -π). En så smuk inversionsformel må kunne ses i et højere perspektiv. Hvis faktoren $1/\sqrt{(x-y)}$ ikke havde været det, havde problemet lydt: løs ligningen

$$g(y) = \int_y^\infty f(x)\, dx.$$

Og denne integralligning kan løses ved blot at differentiere begge sider: f(y) = -g'(y). Disse to integraldannelser er specialtilfælde af følgende integraldannelse: Lad α være et positivt tal, da kan vi for en funktion f(x) definere en ny funktion $I^\alpha f(y)$ ved integralet

$$I^\alpha f(y) = 1/\Gamma(\alpha) \int_y^\infty f(x)\, (x-y)^{\alpha-1}\, dx,$$

hvor Γ(x) er den såkaldte *gammafunktion* (den omtales nærmere i afsnittet om Riemanns primtalsteori), her skal blot siges at for et naturligt tal n er Γ(n) =

(n-1)! og $\Gamma(\frac{1}{2}) = \sqrt{\pi}$. Denne integraltransformation har følgende elegante egenskaber:

$$I^{\alpha}(I^{\beta}f) = I^{\alpha+\beta}f \qquad (I^{\alpha}f)' = I^{\alpha}(f') \qquad (I^{1}f)' = -f \qquad (I^{\alpha}f)' = -I^{\alpha-1}f$$

(den sidste gælder for $\alpha > 1$). Bemærk at den anden siger at operationerne I^{α} og differentation kan ombyttes, og at den tredie (hvor $\alpha = 1$) er den ovennævnte formel $g'(y) = -f(y)$ (idet $\Gamma(1) = 1$), og at den sidste siger at $f' = -I^{-1}f$, *hvis* I^{α} var defineret for $\alpha = -1$, men da den ikke er det, må vi tage I^{α} for $\alpha > 1$ på begge sider af lighedstegnet og anvende den første og anden egenskab.

Vi får Abels teori ved at sætte $\alpha = \frac{1}{2}$, så har vi nemlig at Abels integralligning kan skrives $g = \sqrt{\pi}\, I^{1/2}f$, og så følger af de tre første formler at $I^{1/2}(g') = (I^{1/2}g)' = \sqrt{\pi}\, (I^{1/2}(I^{1/2}f))' = \sqrt{\pi}\, (I^{1}f)' = -\sqrt{\pi}\, f$, og dette er Abels løsningsformel.

Og nu til hvordan Radon omformer Abels inversionsformel til sin egen inversionsformel. Vi betegner materialetæthedsfunktionen f(p). Lad $L(\rho, \theta)$ være den linie som har afstand ρ fra origo og hvis normal har vinkel θ (fra x-aksen), og lad $f\hat{}(\rho, \theta)$ være integralet af f(p) langs $L(\rho, \theta)$ (det tal vi har kaldt "modstanden" i linien):

$$f\hat{}(\rho, \theta) = \int_{L(\rho, \theta)} f(p)\, ds,$$

idet p er det til parameteren s svarende punkt på linien. Og lad for et punkt p' og et tal $r \geq 0$, $M_{p'}f\hat{}(r)$ være middelværdien af $f\hat{}(\rho, \theta)$ for alle tangenterne til cirklen om p' med radius r, dvs.:

$$M_{p'}f\hat{}(r) = 1/(2\pi) \int_0^{2\pi} f\hat{}(\rho, \theta)\, d\theta,$$

hvor ρ er afstanden fra origo af den tangent hvis normal har vinkel θ, dvs.: ρ = r + projektionen af vektoren fra origo til p på retningen θ. For ethvert θ er der to tangenter til cirklen hvis normal har vinkel θ, hvis den ene er knyttet til θ, må den anden knyttes til $\theta+\pi$ (vi kan godt tillade ρ at være negativ).

Radons inversionsformel siger at vi fra denne funktion $M_{p'}f\hat{}(r)$ kommer tilbage til funktionen f(p) ved:

$$f(p') = -1/\pi \int_0^\infty (dM_{p'}f\hat{}(r)/dr)\, dr/r.$$

I definitionen af $M_{p'}f\hat{}(r)$ indgår to integrationer - én langs en ret linie og én langs en cirkel - og Radon viser at disse i en vis forstand kan ombyttes, idet

$$M_{p'}f\hat{}(r) = 2 \int_r^\infty M_{p'}f(r^*)/\sqrt{(1-(r/r^*)^2)}\, dr^*,$$

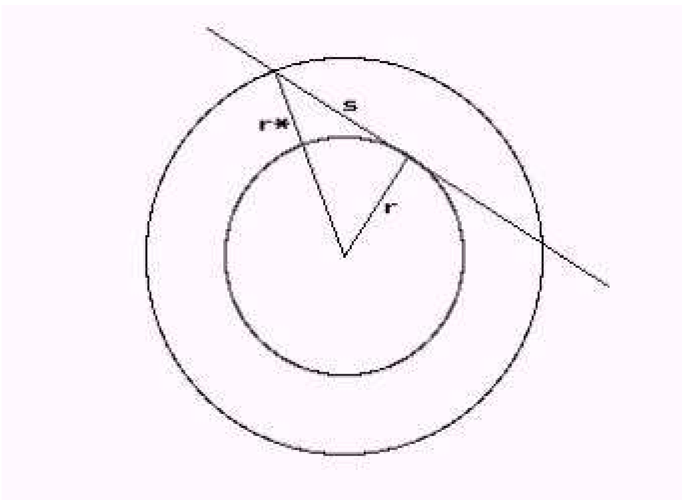
hvor $M_{p'}f(r^*)$ er middelværdien af f(p) langs cirklen C(p', r*) med centrum i p' med radius r*, dvs.

$$M_{p'}f(r^*) = 1/(2\pi r^*) \int_{C(p',\, r^*)} f(p)\, ds.$$

Med figurens betegnelser følger nemlig af $s = \sqrt{(r^{*2} - r^2)}$ at $ds/dr^* = r^*/\sqrt{(r^{*2} - r^2)} = \sqrt{(1-(r/r^*)^2)}$ og dermed at

$$\int_0^\infty f(p)\, ds = \int_r^\infty f(p)/\sqrt{(1-(r/r^*)^2)}\, dr^*$$

(p er punktet svarende til s). Dette integral skal tages i begge retninger af linien og integreres rundt langs den inderste cirkel, herved fås udtrykket for $M_{p'}f\hat{}(r)$.



Men denne sammenhæng imellem $M_{p'}f\hat{}(r)$ og $M_{p'}f(r^*)$ er Abels integralligning: for hvis vi erstatter r med $\sqrt{y}$ og r* med $\sqrt{x}$ har vi at

$$M_{p'}f\hat{}(\sqrt{y}) = \int_y^{\infty} M_{p'}f(\sqrt{x})/\sqrt{(x-y)}\, dx$$

(idet vi har benyttet at $d(\sqrt{x})/dx = dx/(2\sqrt{x})$). Så derfor har vi ifølge Abel at

$$M_{p'}f(\sqrt{x}) = -1/\pi \int_x^{\infty} (d(M_{p'}f\hat{})(\sqrt{y})/dy)/\sqrt{(y-x)}\, dy.$$

Og benytter vi at $d(M_{p'}f\hat{})(\sqrt{y})/dy = (M_{p'}f\hat{})'(\sqrt{y}) \,/(2\sqrt{y})$, erstatter y med r^2 og sætter $x = 0$, så får vi Radons formel (da $M_{p'}f(0) = f(p')$):

$$f(p') = -1/\pi \int_0^{\infty} (dM_{p'}f\hat{}(r)/dr)\, dr/r.$$