

Grundläggande 1D och 2D Signalbehandling för Bilder

*Baserat på ett äldre kompendium av
Per-Erik Danielsson*

Maria Magnusson
Avdelningen för datorseende
Institutionen för systemteknik
Linköpings universitet

Linköping, September 2013



Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Institutionen för systemteknik
Linköpings universitet
581 83 Linköping

Innehåll

1	Introduktion till Bildbehandling	3
1.1	Signalbehandling och digitala bilder	3
1.2	En modell för bildanalys	4
1.3	Datorseende	5
1.4	Bildkodning	5
1.5	Maskinvara för bildbehandling	6
1.6	Bildgenerering	6
2	Endimensionell signalbehandling	9
2.1	Kontinuerlig fouriertransform	10
2.2	Dirac-funktionen	16
2.3	Linjärt tidsinvariant system, impulssvar och faltning	16
2.4	Teorem för fouriertransform	17
2.5	Sampling och rekonstruktion	20
2.6	Fönstring	23
2.7	Tidsdiskret fouriertransform (TDFT)	24
2.8	Diskret fouriertransform (DFT) och cirkulär faltning	24
2.9	Från kontinuerlig till diskret fouriertransform (DFT)	25
2.10	Diskret faltning och faltningskärnor	27
3	Tvådimensionell signalbehandling	33
3.1	2D fouriertransform	33
3.2	2D sampling	37
3.3	2D DFT	41
3.4	Samband: samplad kontinuerlig och samplad diskret bild	42
3.5	2D-samband för separabla funktioner	42
3.6	2D-samband för rotationssymmetriska funktioner	43
3.7	2D faltning och 2D faltningskärnor	43
3.8	2D faltningskärnor för lågpasfiltrering	46
3.9	Faltningkärnor för derivering	49
3.10	Normalisering av faltningskärnor	56
4	Omsampling av bilder	57
4.1	Translation och förstoring (uppsampling)	57
4.2	Rotation	64
4.3	Förminskning (nedsampling)	65
	Litteraturförteckning	69

Tack

Detta kompendium är baserat på ett äldre kompendium av Per-Erik Danielsson. Även andra personer har arbetat med kompendiet under årens lopp, såsom Karin Wall, Ingemar Ragnemalm, Olle Seger, samt jag själv, Maria Magnusson. Under sommaren 2011 överförde Kamalaker Reddy Dadi text och ekvationer till L^AT_EX.

Kapitel 1

Introduktion till Bildbehandling

1.1 Signalbehandling och digitala bilder

En **en-dimensionell** signal är en funktion som endast beror av en variabel. Ett exempel på detta är sinusvågen

$$x(t) = \sin(t). \quad (1.1)$$

På liknande sätt är en **n-dimensionell** signal en funktion som beror av n st oberoende variabler. Ett exempel på en **två-dimensionell** signal är

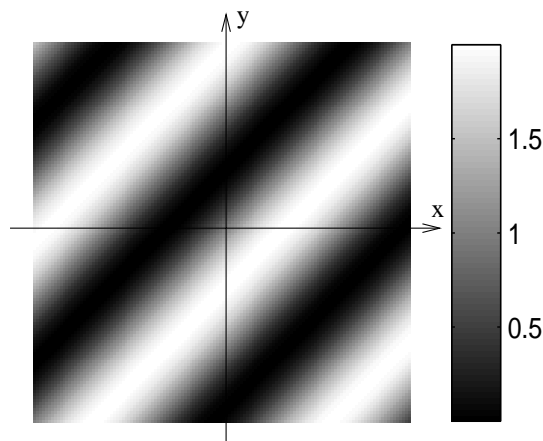
$$f(x, y) = \sin(x - y) + 1. \quad (1.2)$$

Inom **signalbehandlingen** begränsar man sig ofta till en-dimensionella signaler med tiden t som variabel. Två-dimensionell signalbehandling kallas ofta **bildbehandling**. Det vi i dagligt tal menar med en bild är ju en två-dimensionell funktion. De oberoende variablerna betecknas ofta med x och y och betraktas som spatiala (rums-) koordinater. Den två-dimensionella signalen $f(x, y)$ i (1.2) kan illustreras med Fig. 1.1 nedan, där $f(x, y) = 0$ avbildas på svart, $f(x, y) = 2$ avbildas på vitt och övriga värden på $f(x, y)$ med proportionella gråskalevärden.

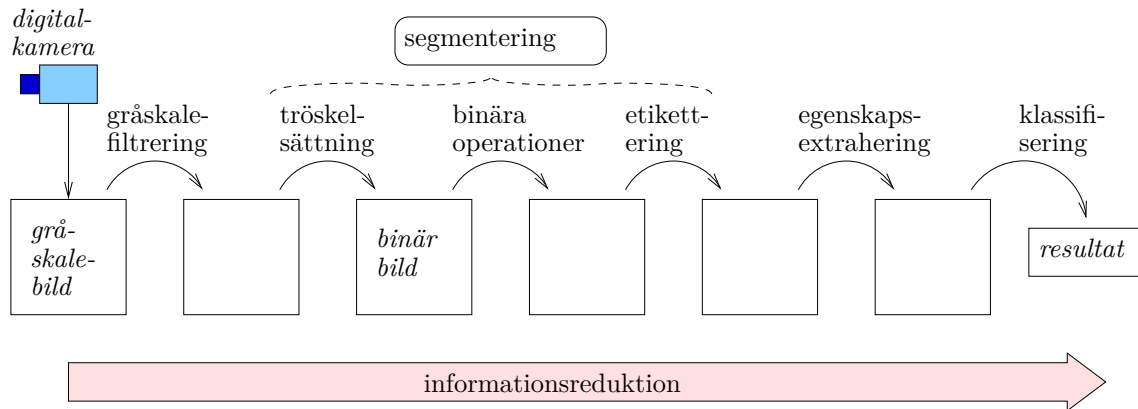
Det bör också nämnas att disciplinen bildbehandling också sysslar med mång-dimensionella bilder. Så behandlas t ex bildsekvenser där tiden t är den tredje variabeln, och bildvolymmer där (x, y, z) är ett vanligt sätt att beteckna de tre rums-koordinaterna.

Generaliseringen från en-dimensionell till fler-dimensionell signalbehandling är i långa stycken rättfram och relativt självklar. Oavsett detta bör utvidgningen till två dimensioner göras och noteras. Till detta kommer att effekter och samband existerar för två-dimensionella signaler som helt saknar motsvarighet i en-dimensionell signalbehandling. Ett sådant fall är rotation.

En **digital bild** $f_D(i, j)$ är en två-dimensionell array med bildpunkter. En bildpunkt brukar kallas pixel (picture element). Antalet pixlar utgör bildens "storlek". En vanligt storlek är $512 \times 512 = 2^{18} =$



Figur 1.1. Gråskalebild enligt (1.2).



Figur 1.2. En bildanalysmodell.

1/4 Mpixel. Oftast kvantiseras pixelvärdet så att ett ändligt antal gråskalevärden kan representeras. Åtta bitar ger t ex 256 gråskalenivåer vilket gör att minnesutrymmet för den ovan nämnda bilden blir 1/4 Mbyte. I en digital bild är alltså de två spatiala koordinaterna i och j **diskretiserade** och amplituden $f_D(i, j)$ är ofta **kvantiserad**. En digital bild $f_D(i, j)$ representerar oftast en fysikalisk bild eller en kontinuerlig två-dimensionell signal $f(x, y)$. En översättning mellan dem kan vara

$$f_D(i, j) = f_D\left(\frac{x}{\Delta} + 257, \frac{y}{\Delta} + 257\right) = \begin{cases} 255, & f(x, y)/k \geq 255, \\ 0, & f(x, y)/k \leq 0, \\ \text{round}[255f(x, y)/k], & \text{annars,} \end{cases} \quad (1.3)$$

där $1 \leq i \leq 512$ och $1 \leq j \leq 512$, dvs bildstorleken är 512×512 , k är en konstant, Δ är samplingsavståndet och amplituden $f_D(i, j)$ är kvantiserad i 256 nivåer från 0 till 255. Om bilden sedan ska visas på en dataskärm, måste de 256 olika nivåerna omvandlas till olika färger eller gråskalevärden. Detta görs via en så kallad färgtabell. Den vanligaste gråskalefärgtabellen avbildar värdet 0 på svart, värdet 255 på vitt och mellanliggande värden proportionellt på olika gråtoner. Numera blir det mer och mer vanligt att fler än 8 bitar används för att lagra varje pixelvärde. I MATLAB t ex, lagras pixlarna oftast i flyttalsformat med dubbel precision, 64 bitar, vilket gör att minnesåtgången för att lagra en bild ökar med en faktor 8. Då bilden ska visas på dataskärmen måste dock flyttalsvärdena först omvandlas till heltalsvärden mellan 0 och 255 (typiska värden) som via färgtabellen konverteras till de färger eller gråtoner som ska visas på skärmen.

Enligt ekvation (1.3) kan den digitala bilden knytas till den bakomliggande kontinuerliga signalen. Samplingsteomet ger oss vägledning att bedöma om omvändningen också är möjlig, dvs om den bakomliggande kontinuerliga signalen kan erhållas ur den digitala bilden. Inom bildbehandlingen arbetar vi med **kontinuerliga och diskreta linjära filter**. Linjära filter definieras och konstrueras antingen i rumsdomänen eller i frekvensdomänen för att uppfylla speciella kriterier för brusundertryckning, kantdetektering, linjedetektering, undertryckande av höga frekvenser eller låga frekvenser eller för framhävande av vissa frekvensband. Efter att ha definierat ett filter så kan den egentliga filtreringen utföras antingen i rumsdomänen såsom faltning eller i frekvensdomänen genom att använda diskret fouriertransform DFT och multiplikation. En minskning av skärpan i bilden (smoothing), kan sålunda göras med en lokalt viktad medelvärdesbildning. Detta är då ett linjärt filter som definieras med hjälp av ett antal viktskoefficienter i rumsdomänen och som utföres genom faltning. Men filtret kan också beskrivas i termer av frekvens och är då ett lågpasfilter.

1.2 En modell för bildanalys

Fig. 1.2 visar en enkel modell för **bildanalys** som innehåller flera olika bildbehandlingsoperationer. Från vänster till höger finner vi först en **digital-kamera** som med hjälp av optik tar in en bild av en 3D-scen, samplar och digitaliserar en bild. Här sker en oerhörd datareduktion. Naturens stokastiska processer har sitt ursprung på atomär nivå och detta gigantiska informationsflöde kan vi endast registrera delvis.

Nästa steg är **gråskalefiltrering**. Det är vanligt att de första stegen i en typisk bildbehandlingssekvens just består av linjär filtrering. Erfarenheten har emellertid visat att mycket få problem kan

lösas med enbart linjär filtrering. I högre grad än för traditionell signalbehandling gäller att bildbehandling tycks erfordra icke-linjära filtreringar och transformationer. Olinjär filtrering innebär att en utgångspixel inte längre kan uttryckas som en linjär viktad summa av ingångspixel. Olinjära filter kan inte heller definieras i frekvensdomänen. En vanlig olinjär filtreringsoperation är att producera medianen (inte medelvärde) av en omgivning av bildpunkter hos ingångsbilden.

En speciell form av olinjär operation är **tröskelsättning**, dvs översättningen av en flernivåbild till en två-nivåers binär bild på så sätt att varje pixel i gråskalebilden jämförs med en tröskel. Efter tröskelsättningen har pixlarna endast två olika värden, 0 och 1, där 0 betyder bakgrund och 1 betyder objekt. Ett antal sammanhängande objekt utgör ett objekt. På denna binära bild kan man nu utföra **binära operationer**. Det finns många olika varianter, varav en är att fylla igen hål i objekt. Vid **etiketteringen** tilldelas pixlarna i de olika objekten olika etiketter så att man ska kunna skilja dem åt. Att urskilja och etikettera ett antal objekt från bakgrunden kallas **segmentering**. Det finns även andra metoder för segmentering än tröskelsättning - binärfiltrering - etikettering som visas i figuren.

I nästa steg kan det bli aktuellt att för ett visst segmenterat objekt ta fram mätvärden av typ medelintensitet, textur etc. **Egenskapsextraktionen** resulterar i en serie mätvärden för varje objekt i bilden. Vilka mätvärden (features) som man bör räkna fram ur bilden är synnerligen tillämpningsberoende, men som exempel kan vi nämna storlek (area), form (kan definieras på många sätt), färg, orientering, medelintensitet, granularitet, position dvs (x, y) -lägen i bilden, närmsta avstånd till annat objekt, etc. Allt detta samlas i en egenskapslista. Resultatet efter dataextraktionen är inte längre en bild utan en lista. Flera algoritmer har utvecklats för segmentering och egenskapsextraktion. Notera att den ursprungliga digitala bilden, liksom producerade mellanresultat inte behöver kastas bort under segmentering och egenskapsextraktion.

I det sista steget användes egenskapslistan för att **klassificera**, dvs artbestämma, känna igen, avgöra vilken sort ett objekt tillhör. I vissa tillämpningar kan det vara hela bilden eller scenen som skall klassas. I ganska många fall, t ex industriell avsyning, screening av medicinska preparat, finns bara två möjliga svar; godkänd/icke godkänd, cancer/icke cancer. Oftast bör man införa en tredje klass, vet ej, dvs undvika beslut i vissa fall, vilket komplicerar klassningsalgoritmerna. I många fall är det emellertid mycket vunnit att göra ett binärt beslut med innebörden godkänd/tveksam. Om det godkända fallen blir avsevärt flera än de tveksamma är automatiseringen meningsfull även om de tveksamma måste lämnas till icke-automatiserad eftergranskning. Klassificering eller mönsterigenkänning, som det också kallas, är en väl utvecklad disciplin som av tradition baseras på matematiskt-statistiska metoder. En viktig och teoretiskt intressant aspekt på klassificering- och igenkänningsalgoritmer är möjligheten att träna fram ett optimalt beteende (beslutsfattande). Användning av neuronät har i hög utsträckning handlat om att uppnå läraktighet i denna mening. Detta kompendium behandlar inte klassificering vidare.

1.3 Datorseende

Datorseende kan sägas innefatta allt om grundläggande bildbehandling, men utökat med 3D-geometri, kamerakalibrering och objektigenkänning. En pågående trend är att koppla samman datorseende med området maskininlärning. Då skapar man system som analyserar och lär sig från bilder. Dessa förvärvade kunskaper utnyttjas sedan för att lösa nya uppgifter.

1.4 Bildkodning

I samband med bildbehandling bör några ord sägas om området **bildkodning**. Det praktiska målet för kodning är att komprimera data så att bilden kan lagras och överföras mera effektivt. Många bilder är i själva verket högeligen redundanta; ofta kan datamängden reduceras (komprimeras) med en faktor 10. Ett typiskt exempel är ett dokument, dvs en svart-vit binär bild, som ofta innehåller ganska stora areor av tomma vita ytor. I stället för att spendera en bit på varje bildpunkt så kan man producera en kodad version på så sätt att varje linje innehåller de x -lägen där bildens värde växlar från svart till vitt eller tvärtom. Ännu bättre är måhända att endast registrera avstånden mellan två sådana omslag i bildpunktsvärdet vilket kallas run-length-coding.

En kodares prestanda mäts i form av komprimeringsfaktorn för en given maximal distorsion, dvs den tillåtna differensen mellan originalet och den rekonstruerade, den avkodade bilden. Under arbetet med att finna den viktiga informationen i en bild och undertrycka den icke viktiga så kommer bildkodning att använda liknande filter och filtreringsmetoder som de som används i bildbehandling.

Ovanstående indikerar att lågnivåbildbehandling och bildkodning ofta är identiska vad gäller både

mål och metoder. En skillnad kan dock vara att bildkodning normalt avser att hantera alla slags bilder medan bildbehandling/bildanalys vanligtvis avstämmer till en speciell applikation där endast en mycket begränsad typ av bilder kan förekomma.

1.5 Maskinvara för bildbehandling

Maskinvara för bildbehandling är vanligtvis men inte alltid åtskild från det bildgenererande organet. För de följande diskussionerna antar vi emellertid att bilden har producerats och digitaliserats. En fruktbar diskussion kräver ytterligare några distinktioner, nämligen mellan

- lågnivåbildbehandling - högnivåbildbehandling
- speciell maskinvara - generell maskinvara
- utrustning för slutanvändare - utrustning för utveckling och forskning

Eftersom så stor del av lågnivåbildbehandling tycks bestå av likformiga och ganska enkla operationssteg, t ex omgivningsoperationer, har denna del av bildbehandlingstekniken varit ett speciellt mål för ny datorarkitektur, t ex parallella processorer i olika variationer och organisationsformer.

Det bör understrykas att bildbehandlingstillämpningar (om de är lyckosamma) rör sig gradvis från experimenterande med en algoritm eller metod på ett fåtal bilder över programkörningar över ett större material för att till slut bli mera produktiva exekveringar för slutanvändaren. Hastighetskravet finns här hela tiden. Försök och misstag är typiska i de första stegen av proceduren där användaren vill ha snabbt svar för att kunna utvärdera sin egen tankegång. Redan på detta stadium kan algoritm innebära ganska krävande beräkningar. Behovet av snabbhet vid mera produktiva körningar är självklart.

Hastighetskravet kommer att öka i framtida tillämpningar. Sant tredimensionella bilder, volymer, produceras av många nya bildgenererande organ. Nya satelliter kommer att förse oss med en aldrig sinande ström av bilder av långt högre upplösning än vad vi har idag. Dessutom kommer användning av generella metoder för datorseende att öka snarare än minska behovet av beräkningskraft för att processa en viss bild. Detta gör att trots att datorerna hela tiden blir snabbare och snabbare så är det ändå viktigt att bildbehandlingsoperationerna implementeras effektivt. Det är därför som vi i kompendiet ibland kommer med förslag på hur de olika algoritmerna ska kunna implementeras på ett effektivt sätt.

1.6 Bildgenerering

Med bildgenerering (engelska: Imaging) menar vi här skapandet, genererandet av bilder. Detta är alltså en lätt avvikelse från huvudområdet bildbehandling. Emellertid vill vi betona att bildbehandling i nästan alla tillämpningar är starkt kopplad till den process som genererar själva bilden. Dessutom anser vi att det är lämpligt att här peka på de enastående framsteg som har ägt rum inom detta område. En tänkbar indelning av bildgenererande organ och teknologi skulle kunna vara följande:

1) Bildgenerering med elektromagnetisk strålning

- röntgenkristallografi
- medicinsk och industriell röntgen
- röntgendatortomografi (engelska: Computer Tomography (CT))
- gamma-kamera
- SPECT (Single Photon Emission CT)
- optik och optiska sensorer av alla slag
- CCD- och CMOS-sensorer, fotomultiplikatorer
- laserscannern för djupmätning
- mikroskop, teleskop
- bildsensorer i satelliter

- IR-sensorer och IR-kameror
- mikrovågsdetektorer och antenn
- syntetisk aperturradar, radioastronomi
- MRI (Magnetic Resonance Imaging)

2) Bildgenerering med partikelstråle

- transmissionselektronmikroskopi (engelska: transmission electron microscopy (TEM))
- svepelektronmikroskopi (engelska: scanning electron microscopy (SEM))
- positronkamera (engelska: positron emission tomography (PET))
- bildgenerering inom kärnfysikalisk forskning

3) Bildgenerering med ultraljud vilket inkluderar ett stort antal olika varianter, mestadels för medicinskt bruk.

Praktiskt taget alla dessa olika tekniker är i ett tillstånd av mycket snabb utveckling. Även det väl utvecklade området ljusmikroskopi ändrar sig snabbt under trycket av eventuellt datoriserad bildbehandling. Datorstyrda mikroskopbord, automatisk fokusering, tvåkamera-system för hög och låg upplösning, zoomningsmöjligheter, laserscanningteknik, konfokala system är några metoder som kan adderas till äldre av typen ljusfält, mörkfält, faskontrast, fluorocens etc.

Kapitel 2

Endimensionell signalbehandling

Idag sker signalbehandlingen i elektrisk apparatur huvudsakligen digitalt. Digitala signaler är samplade, vilket innebär att man läser av signalvärden på en kontinuerlig signal med ett visst tidsmellanrum. Man måste också kvantisera signalens amplitud, vilket ger en tids- och amplituddiskret signal. En sådan signal kallas **digital**, medan en signal som är tids- och amplitudkontinuerlig kallas **analog**. Mobiltelefoner, i stor utsträckning vanlig telefoni, CD- och DVD-skivor, ljudspelare och olika styrsystem är exempel på tillämpningar som använder sig av digitala signaler. Det är ofta utveckling av billiga och små digitala enheter som möjliggör en övergång från analog till digital teknik.

Trots att mycket av den signalbehandling som utförs idag sker digitalt på digitala signaler, så kan man inte bortse från den bakomliggande kontinuerliga signalen. I telefoni och ljudanläggningar exempelvis, är det nödvändigt att återskapa den analoga signalen ur den digitala versionen. Den analoga elektriska signalen skickas till en högtalare där den omvandlas till ljud så att vi människor kan höra den. Vi kommer längre fram i kompendiet, när vi studerar sampling, se att det finns en mycket stark koppling mellan den bakomliggande kontinuerliga signalen och den diskreta samplade versionen.

Detta kapitel ger en kortfattad genomgång av teorin kring 1D signalbehandling. För den läsare som vill fördjupa sig i området rekommenderas böckerna [1], [2], [6], [9], [10] och [12] av Bracewell, Cha, Oppenheim, Söderkvist och Svärdström.

2.1 Kontinuerlig fouriertransform

Den en-dimensionella fouriertransformen av en signal $g(t)$ definieras

$$G(\omega) = \int_{-\infty}^{\infty} g(t)e^{-j\omega t} dt \quad (2.1)$$

och dess invers är

$$g(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega)e^{j\omega t} d\omega \quad (2.2)$$

Det är vanligt att man betecknar fouriertransformen av en funktion med samma bokstav som funktionen, fast man använder stor bokstav. Ekvation (2.1) ovan är en funktion av vinkelfrekvensen ω . Om man utför variabelbytet $\omega = 2\pi f$ så fås följande alternativa matematiska beskrivning av fouriertransformen och dess invers,

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-j2\pi ft} dt, \quad (2.3)$$

$$g(t) = \int_{-\infty}^{\infty} G(f)e^{j2\pi ft} df. \quad (2.4)$$

Ibland betecknas fouriertransformen $G(f) = \mathcal{F}[g(t)]$ och den inversa fouriertransformen på motsvarande sätt $g(t) = \mathcal{F}^{-1}[G(f)]$.

Alla signaler som existerar i naturen och som vi kan mäta är reella. För en reell funktion $g(t)$ kan man visa att dess fouriertransform $G(f)$ är **hermitisk**, dvs

$$G(f) = G^*(-f). \quad (2.5)$$

Funktionerna $G(f)$ och $G(-f)$ är alltså varandras konjugat. Detta kan också uttryckas som

$$\operatorname{Re}G(f) = \operatorname{Re}G(-f), \quad (2.6)$$

$$\operatorname{Im}G(f) = -\operatorname{Im}G(-f). \quad (2.7)$$

$\operatorname{Re}G(f)$ är alltså en jämn funktion och $\operatorname{Im}G(f)$ är en udda funktion. Eftersom $G(f)$ är en komplex funktion kan den skrivas på polär form

$$G(f) = A(f)e^{j\Phi(f)}, \quad (2.8)$$

där

$$A(f) = |G(f)| \quad (2.9)$$

kallas för signalens **amplitudspektrum** och

$$\Phi(f) = \arg G(f) \quad (2.10)$$

kallas för signalens **fasspektrum**. Kvantiteten $|G(f)|^2$ kallas för signalens **effekt- eller energispektrum**. I Fig. 2.1 visas den reella signalen $g(t) = \Pi((t-1)/2)$, dess fouriertransform med realdelen $\operatorname{Re}G(f)$ och imaginärdelen $\operatorname{Im}G(f)$. Vidare visas amplitudspektrum $A(f) = |G(f)|$ och fasspektrum $\Phi(f)$.

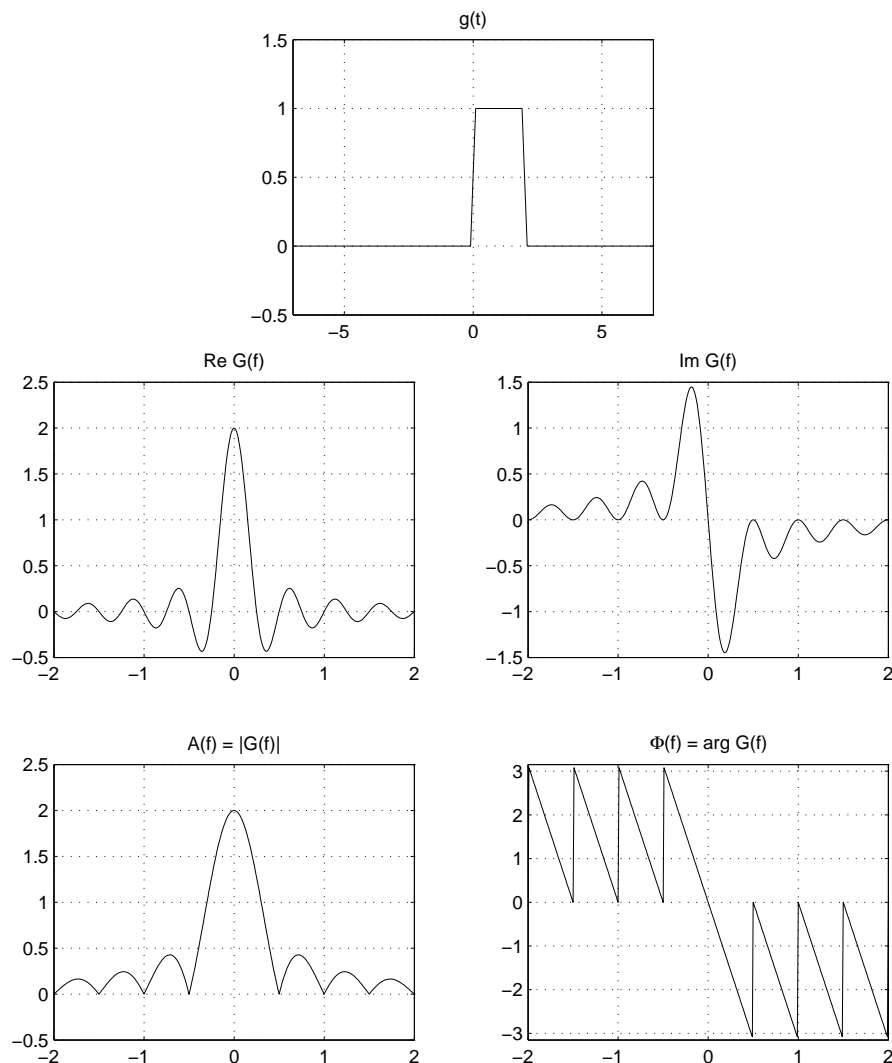
En god hjälp för att intuitivt förstå fouriertransformen är korrespondensen mellan egenskaperna jämn/udda respektive reell/imaginär i signal- och fourierdomänerna. De flesta av dessa korrespondenser blir uppenbara om vi skriver funktionen

$$g(t) = e(t) + o(t),$$

där $e(t)$ är en jämn (even) funktion och $o(t)$ är en udda (odd) funktion,

$$e(t) = 1/2[g(t) + g(-t)],$$

$$o(t) = 1/2[g(t) - g(-t)].$$



Figur 2.1. Signalen $g(t) = \Pi((t-1)/2)$, dess fouriertransform $\text{Re}G(f) + j\text{Im}G(f)$, amplitudspektrum $A(f) = |G(f)|$ och fasspektrum $\Phi(f) = \arg G(f)$.

En jämn funktion $e(t)$ består av enbart cosinuskomponenter, en udda funktion $o(t)$ av enbart sinuskomponenter. Av detta följer att fourierformen kan skrivas

$$G(f) = 2 \int_0^{\infty} e(t) \cos(2\pi ft) dt - 2j \int_0^{\infty} o(t) \sin(2\pi ft) dt. \quad (2.11)$$

Om ovanstående resultat (2.11) kombineras med det faktum att “fouriertransformen av en reell funktion är en hermitisk funktion” (2.5) fås de viktiga observationerna “fouriertransformen av en jämn, reell funktion är jämn och reell” och “fouriertransformen av en udda, reell funktion är udda och imaginär”. Dessa resultat m fl är illustrerade i Fig. 2.2.

I Fig. 2.3 och Fig. 2.4 visar vi ett antal fourierpar för några vanligt förekommande funktioner. **Impulståget** definieras enligt

$$\frac{1}{T} \text{III} \left(\frac{t}{T} \right) = \sum_{n=-\infty}^{+\infty} \delta(t - nT). \quad (2.12)$$

där $\delta()$ är den så kallade dirac-impulsen (se nedan). För impulstågets fouriertransform gäller

$$\mathcal{F} \left[\frac{1}{T} \text{III} \left(\frac{t}{T} \right) \right] = \text{III}(fT) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta \left(f - \frac{k}{T} \right) \quad (2.13)$$

Rektangelfunktionen har definitionen

$$\Pi(t) = \begin{cases} 0, & |t| > \frac{1}{2}, \\ \frac{1}{2}, & |t| = \frac{1}{2}, \\ 1, & |t| < \frac{1}{2}. \end{cases} \quad (2.14)$$

Triangel-funktionen har definitionen

$$\Lambda(t) = \begin{cases} 1 - |t|, & |t| \leq 1, \\ 0, & |t| > 1. \end{cases} \quad (2.15)$$

Signum-funktionen har definitionen

$$\text{sgn}(t) = \begin{cases} -1, & t < 0, \\ 1, & t > 0. \end{cases} \quad (2.16)$$

Sinc-funktionen har definitionen

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}. \quad (2.17)$$

Jämna impuls-paret har definitionen

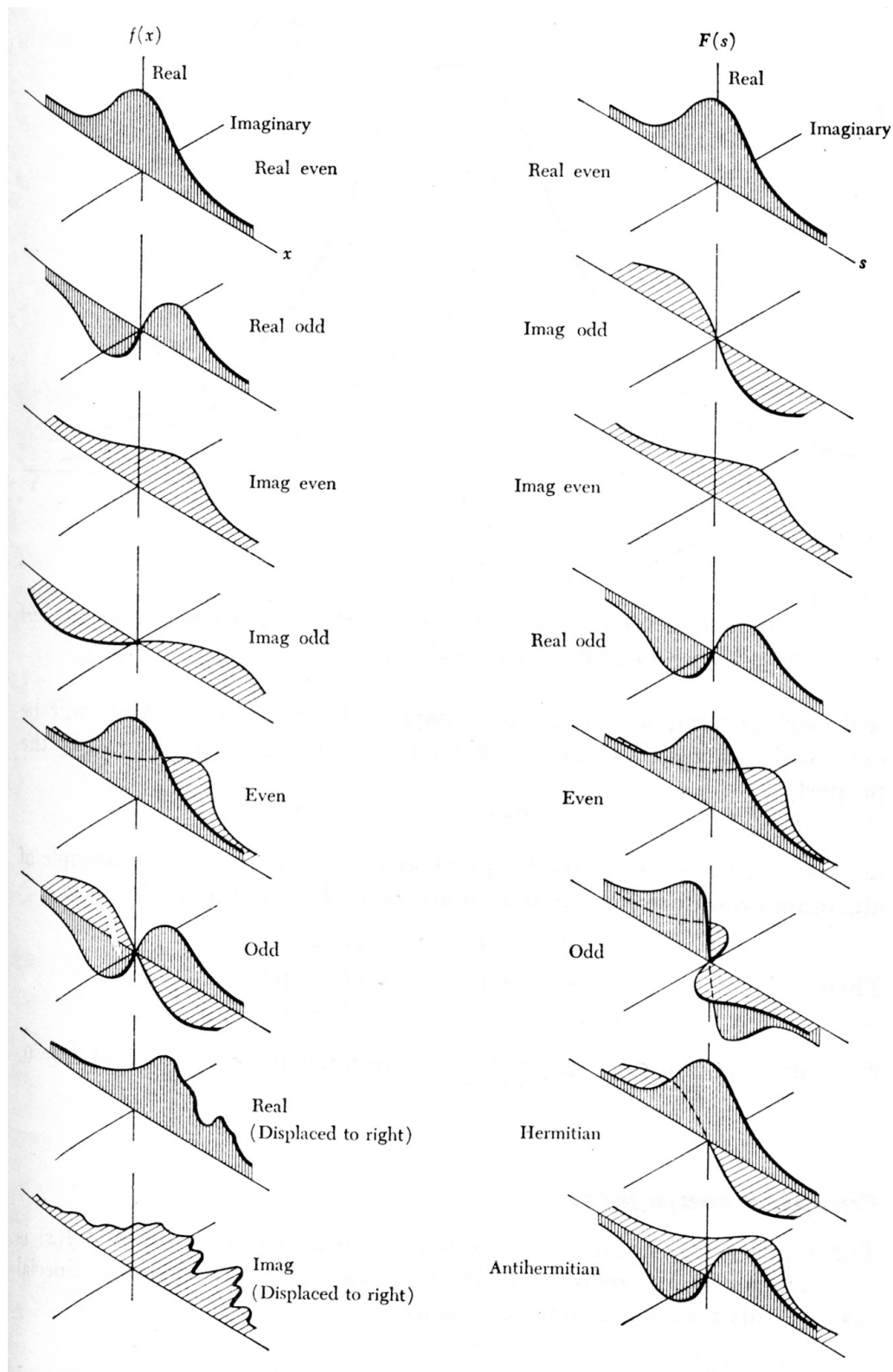
$${}_{\text{I I}}(t) = \frac{1}{2}\delta\left(t + \frac{1}{2}\right) + \frac{1}{2}\delta\left(t - \frac{1}{2}\right) \quad (2.18)$$

och udda impuls-paret har definitionen

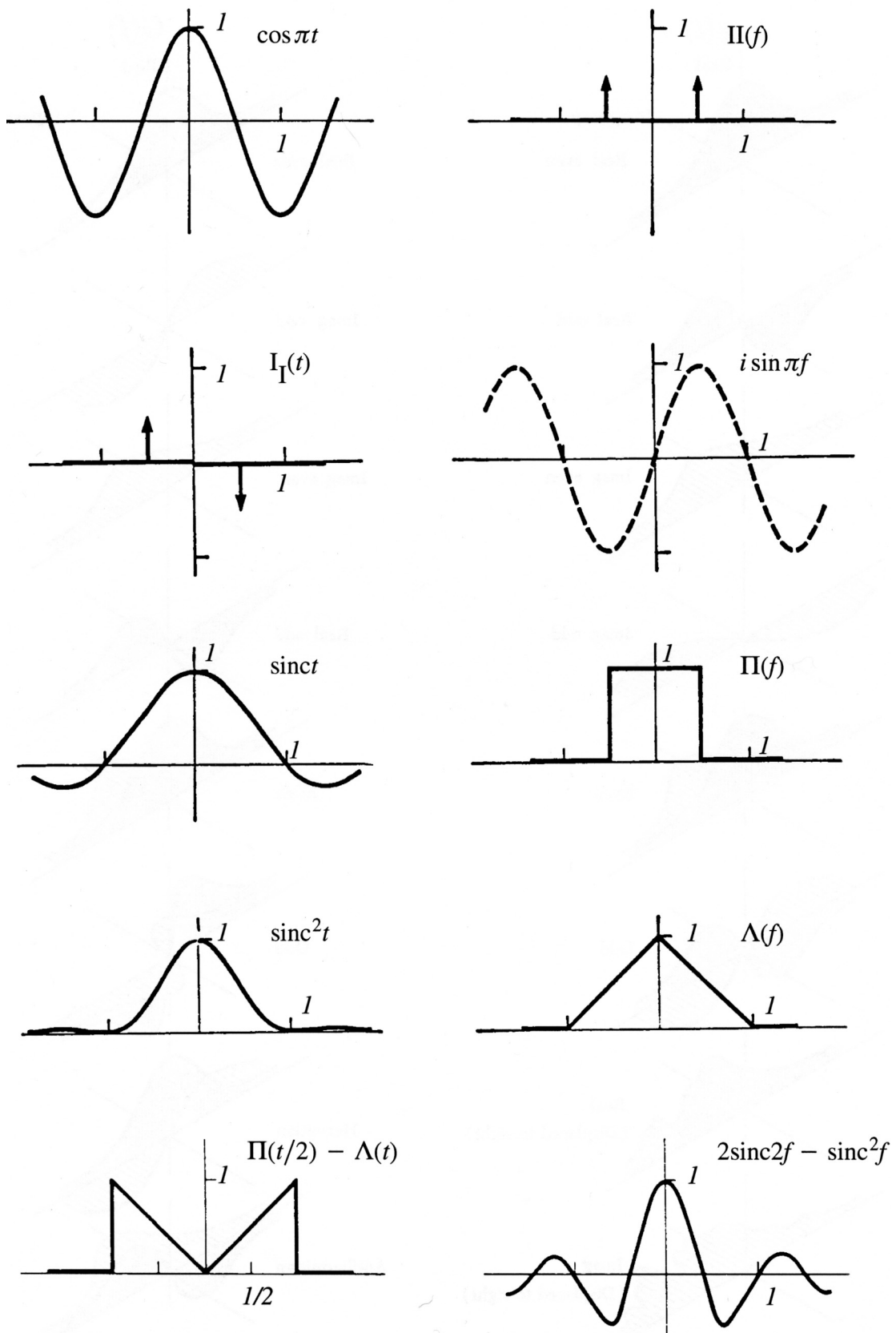
$${}_{\text{I I}}(t) = \frac{1}{2}\delta\left(t + \frac{1}{2}\right) - \frac{1}{2}\delta\left(t - \frac{1}{2}\right). \quad (2.19)$$

Några allmänna observationer är följande:

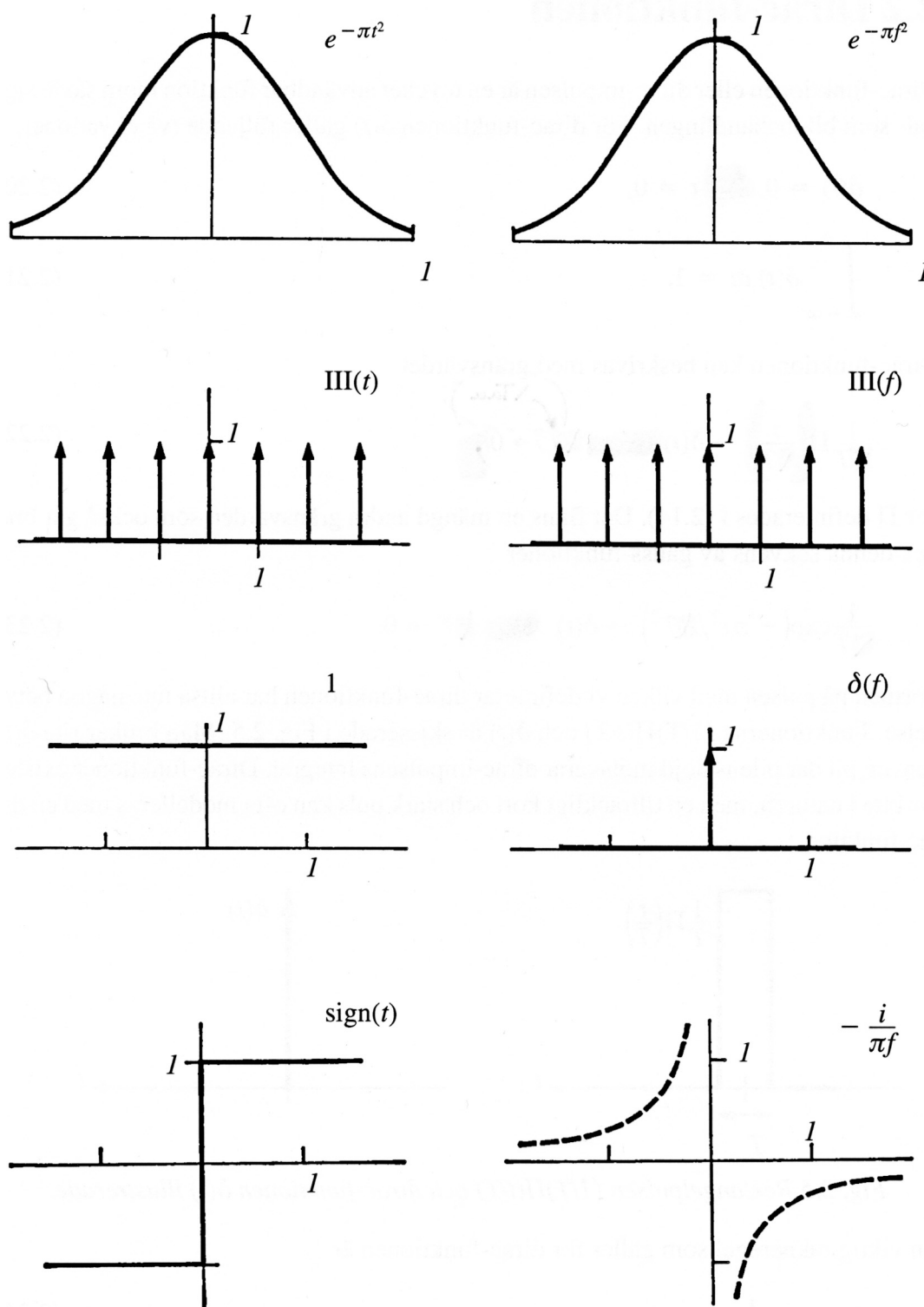
- Diskontinuiteter i en funktion och/eller dess derivator ger långsam avklingning till 0 av transformen för $f \rightarrow \infty$. Mera exakt gäller att om n:te derivatan (funktionen själv räknas som 0:te derivatan) är diskontinuerlig så avtar $G(f)$ som $\mathcal{O}(f^{-(n+1)})$.
- Liten utbredning i den ena domänen ger stor utbredning i den andra.
- Ett litet antal funktioner har samma principiella utseende i båda domänerna, däribland den Gaussiska e^{-t^2} och impulståget $(1/T)\text{III}(t/T)$.



Figur 2.2. Fourierpar för jämna och udda funktioner (efter Bracewell).



Figur 2.3. Fourierpar (efter Bracewell).



Figur 2.4. Fortsättning av fourierpar (efter Bracewell).

2.2 Dirac-funktionen

Dirac-funktionen eller dirac-impulsen är en mycket användbar funktion inom såväl signal- som bildbehandling. För dirac-funktionen $\delta(t)$ gäller följande två ekvationer,

$$\delta(t) = 0, \quad t \neq 0, \quad (2.20)$$

$$\int_{-\infty}^{\infty} \delta(t) dt = 1. \quad (2.21)$$

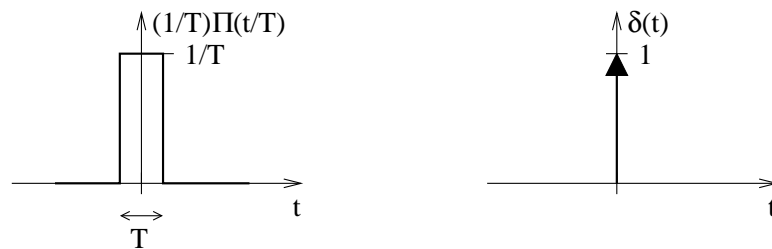
Dirac-funktionen kan beskrivas med gränsvärdet

$$\frac{1}{T}\Pi\left(\frac{t}{T}\right) \rightarrow \delta(t) \quad \text{då } T \rightarrow 0, \quad (2.22)$$

där Π definierades i (2.14). Det finns en mängd andra gränsvärden som också går bra, t ex denna sekvens av gauss-funktioner

$$\frac{1}{T} \exp(-\pi t^2/T^2) \rightarrow \delta(t) \quad \text{då } T \rightarrow 0. \quad (2.23)$$

Formen på pulsen med vilken vi definierar dirac-funktionen har alltså inte någon betydelse. Funktionerna $(1/T)\Pi(t/T)$ och $\delta(t)$ är skisserade i Fig. 2.5. Man brukar rita $\delta(t)$ som en pil där pilens höjd motsvarar dirac-impulsens integral. Dirac-funktioner existerar inte i naturen, men en tillräckligt kort och stark puls kan ofta modelleras med en dirac-funktion.



Figur 2.5. Rektangelpulsen $(1/T)\Pi(t/T)$ och dirac-funktionen $\delta(t)$ illustrerade.

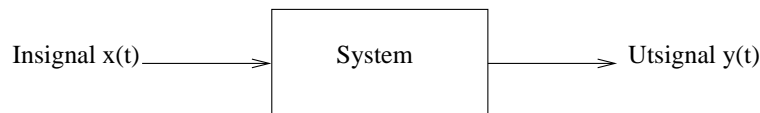
En viktig räkneregel som gäller för dirac-funktionen är

$$\delta(at) = \frac{1}{|a|}\delta(t), \quad (2.24)$$

där a är en konstant och $a \neq 0$. Ovanstående samband inses genom att både HL och VL är 0 för $t \neq 0$ samt att integralen över HL respektive VL är lika med värdet $1/|a|$, vilket inses via (2.22).

2.3 Linjärt tidsinvariant system, impulssvar och faltning

Fig. 2.6 illustrerar innebörden av begreppet **system**. Då systemet påverkas av en insignal $x(t)$ genererar det utsignalen $y(t)$.



Figur 2.6. Ett system som påverkas av en insignal $x(t)$ genererar en utsignal $y(t)$.

Signaler och system är alltså mycket nära knutna till varandra. Ett vanligt sätt att karaktärisera ett system är genom att ange dess **impulssvar**. Låt en dirac-funktion vara insignal till ett system. Utsignalen kommer då att vara systemets impulssvar, vilket ofta anges $h(t)$. Det är vanligt att skriva $h(t)$ på rektangeln som betecknar systemet, se Fig. 2.7.

För ett **linjärt tidsinvariant system (LTI-system)** gäller följande egenskaper:

Linjäritet

Antag att insignalerna $x_1(t)$ och $x_2(t)$ ger utsignalerna $y_1(t)$ och $y_2(t)$. Om då insignalen $A \cdot x_1(t) + B \cdot x_2(t)$ ger utsignalen $A \cdot y_1(t) + B \cdot y_2(t)$ så är systemet linjärt.

Tidsinvarians

Antag att insignalen $x(t)$ ger utsignalen $y(t)$. Om då insignalen $x(t + T)$ ger utsignalen $y(t + T)$ är systemet tidsinvariant.



Figur 2.7. Impulssvaret $h(t)$ illustrerat.

Om impulssvaret $h(t)$ är känt för ett linjärt tidsinvariant system så kan utsignalen $y(t)$ beräknas för en godtycklig kontinuerlig insignal $x(t)$ genom **faltning** enligt formeln

$$y(t) = (x * h)(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau. \quad (2.25)$$

Följdaktligen gäller att faltning med med dirac-funktion inte påverkar originalfunktionen, dvs

$$\delta(t) * h(t) = \int_{-\infty}^{\infty} \delta(\tau)h(t - \tau)d\tau = h(t). \quad (2.26)$$

För faltning använder vi oftast skrivsättet $y(t) = (x * h)(t)$ för att poängtera att faltning inte är en punktvis operation som t ex multiplikation av två funktioner. Ibland kan det dock vara praktiskt att i stället skriva $y(t) = x(t) * h(t)$. Genom ett enkelt variabelbyte kan faltningsintegralen i (2.25) skrivas om till

$$y(t) = (h * x)(t) = h(t) * x(t) = \int_{-\infty}^{\infty} x(t - \tau)h(\tau)d\tau, \quad (2.27)$$

vilket ibland kan vara ett fördelaktigare skrivsätt. Faltning är alltså en kommutativ operation. Faltning enligt (2.27) illustreras i Fig. 2.8. Den vikta versionen av signalen x förskjuts över signalen h . I varje läge beräknas integralen av produkten, vilkens resultat avsätts på platsen för markören. Markören följer med förskjutningen av $x(t - \tau)$ och genomlöper därmed alla resultat-signalens positioner.

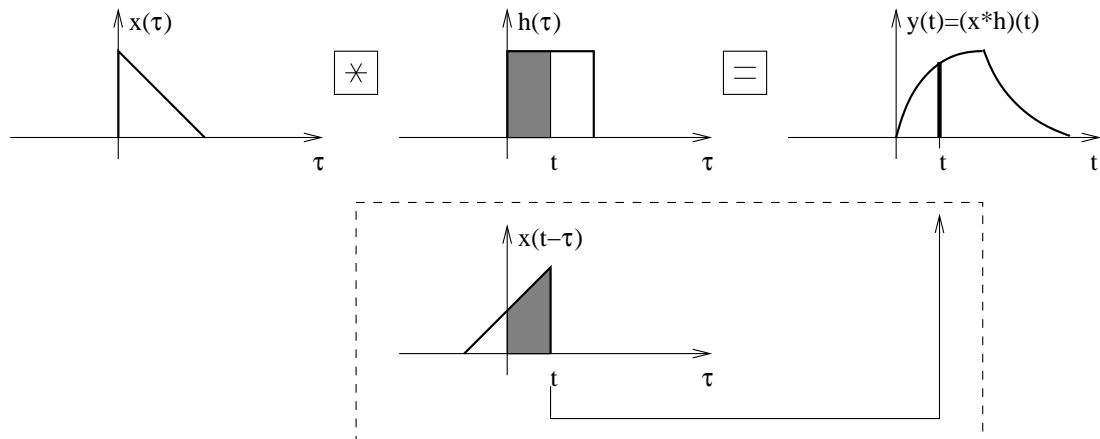
Det är viktigt att känna till vad som händer då en funktion faltas med en förskjuten dirac-funktion. Det gäller att

$$\delta(t - a) * x(t) = x(t - a) \quad (2.28)$$

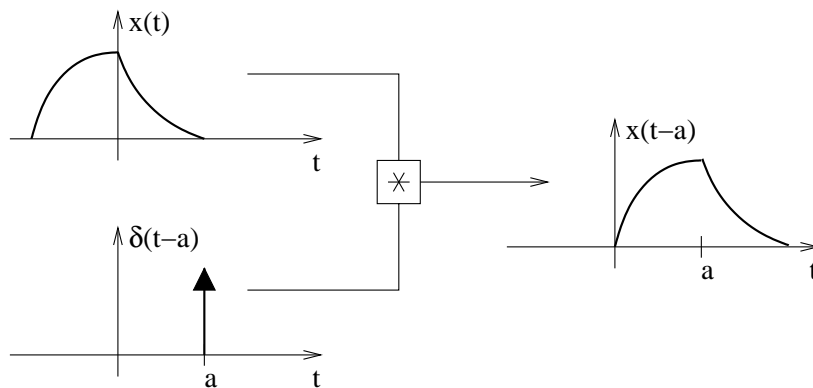
vilket kan formuleras som "Vid faltning mellan en förskjuten dirac-impuls och en funktion, flyttas funktionen till dirac-impulsens läge". Sambandet är också illustrerat i Fig. 2.9.

2.4 Teorem för fouriertransform

För fouriertransformen gäller ett antal grundläggande teorem som vi härmed rekapitulerar utan bevis. Vi underförstår att $\mathcal{F}[g(t)] = G(f)$. Funktionen står till vänster om dubbelpilen och fouriertransformen till höger i teoremen nedan.



Figur 2.8. Faltning av kontinuerliga signaler.



Figur 2.9. Vid faltning mellan en förskjuten dirac-impuls och en funktion, flyttas funktionen till dirac-impulsens läge.

Skalningsteoremet

$$g(at) \Leftrightarrow \frac{1}{|a|} G\left(\frac{f}{a}\right) \quad (2.29)$$

Vi noterar alltså att om funktionen $g(t)$ komprimeras utefter t -axeln med ökande a i signaldomänen så breddas fouriertransformen. Faktorn $1/|a|$ är nödvändig som skalfaktor för bevarande av energin (se Rayleigh's teorem nedan).

Linjäritetsteoremet

$$a \cdot g_1(t) + b \cdot g_2(t) \Leftrightarrow a \cdot G_1(f) + b \cdot G_2(f) \quad (2.30)$$

Translationsteoremet

$$g(t - a) \Leftrightarrow e^{-j2\pi a f} G(f) \quad (2.31)$$

Vid translation i signaldomänen förändras alltså inte amplituden $|G(f)|$ av en viss frekvenskomponent, utan endast fasläget, dvs förhållandet mellan real - och imaginärdel.

Faltningsteoremet

$$(g_1 * g_2)(t) \Leftrightarrow G_1(f) \cdot G_2(f) \quad (2.32)$$

Multiplikationsteoremet

$$(g_1 \cdot g_2)(t) \Leftrightarrow G_1(f) * G_2(f) \quad (2.33)$$

Modulationsteoremet

$$g(t) \cdot \cos(2\pi f_1 t) \Leftrightarrow \frac{1}{2}G(f - f_1) + \frac{1}{2}G(f + f_1) \quad (2.34)$$

Korrelation

$$g_1 \square g_2(t) = g_1(t) * g_2^*(-t) \Leftrightarrow G_1^*(f) \cdot G_2(f) \quad (2.35)$$

Korrelation, betecknad med \square , är detsamma som "faltning utan vikning", dvs om $g_1(t)$ och $g_2(t)$ är reella gäller att $g_1 \square g_2(t) = g_1(-t) * g_2(t)$. Om funktionerna är komplexa tillkommer konjugering.

Autokorrelation

$$g \square g(t) = g(t) * g^*(-t) \Leftrightarrow G^*(f) \cdot G(f) = |G(f)|^2 \quad (2.36)$$

Autokorrelationen är en funktions korrelation med sig själv. Autokorrelationens fouriertransform är alltså funktionens effekt- eller energispektrum.

Derivatateoremet

$$\frac{d}{dt}[g(t)] = g'(t) \Leftrightarrow j2\pi f \cdot G(f) \quad (2.37)$$

Eftersom $\mathcal{F}[g(t)] = G(f)$ kan vi helt enkelt skriva

$$\frac{d}{dt} \Leftrightarrow j2\pi f. \quad (2.38)$$

Derivering av faltning

$$\frac{d}{dt}(g_1 * g_2)(t) = (g_1' * g_2)(t) = (g_1 * g_2')(t) \quad (2.39)$$

Detta inses lättast genom att observera att deriveringsoperatoren är en funktion som enligt närmast föregående teorem har fouriertransformen $j2\pi f$. Ur signalbehandlingssynpunkt är deriveringsoperationen en faltning. Uttrycket ovan är alltså liktydigt med tre funktioner faltade med varandra. Ordningsföljden är likgiltig eftersom faltning är en kommutativ operation. Att derivering är faltning kan inte nog understrykas. Ekvation (2.39) skulle alltså kunna skrivas

$$\frac{d}{dt} * g_1 * g_2 = \frac{dg_1}{dt} * g_2 = g_1 * \frac{dg_2}{dt}.$$

Parsevals teorem (Effektteoremet)

$$\int_{-\infty}^{\infty} g_1(t) \cdot g_2^*(t) dt = \int_{-\infty}^{\infty} G_1(f) \cdot G_2^*(f) df \quad (2.40)$$

Detta kan skrivas

$$\int_{-\infty}^{\infty} g_1(t) \cdot g_2(-t) dt = \int_{-\infty}^{\infty} G_1(f) \cdot G_2(f) df \quad (2.41)$$

om $g_2(t)$ är reell. Om vi låter t vara tid, $g_1(t)$ och $g_2(t)$ ström respektive spänning så utsäger teoremet att den elektriska medeleffekten kan beräknas som summan av ström \cdot spänning för varje frekvens.

Specialfall av Parsevals teorem (Rayleigh's teorem)

$$\int_{-\infty}^{\infty} |g(t)|^2 dt = \int_{-\infty}^{\infty} |G(f)|^2 df \quad (2.42)$$

Om g_1 och g_2 sättes lika i Parsevals teorem enligt (2.40) ovan fås detta specialfall som ibland brukar kallas Rayleigh's teorem. Teoremet utsäger att den totala energin är densamma i signaldomän och fourierdomän.

2.5 Sampling och rekonstruktion

En **samplad** signal $s(t)$ är en funktion som är $\neq 0$ endast i ett antal ekvidistanta punkter på t -axeln. Den representerar den kontinuerliga signalen $g(t)$ genom att

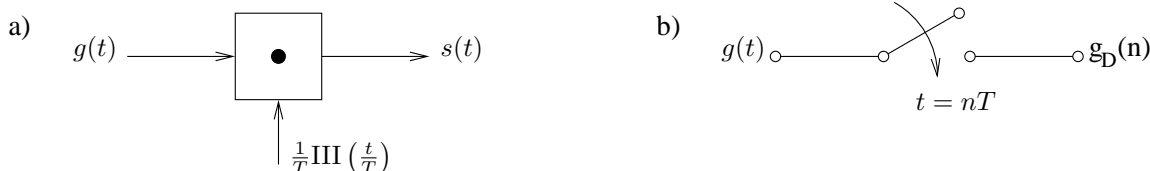
$$s(t) = \begin{cases} g(t) \delta(t - nT), & \text{för } t = nT, \quad n = \dots, -1, 0, 1, 2, \dots \\ 0 & \text{för } t \neq nT. \end{cases}$$

Funktionen $g(t)$ är alltså den "bakomliggande" kontinuerliga signalen som vi samplat (läst av, registrerat, detekterat) med intervallet T utefter t -axeln. Den samplade signalen kan ses som produkten av $g(t)$, $1/T$ och impulståget. Uttrycket för den samplade signalen blir därmed

$$s(t) = g(t) \cdot \frac{1}{T} \text{III} \left(\frac{t}{T} \right) = g(t) \sum_{n=-\infty}^{\infty} \delta(t - nT), \quad (2.43)$$

vilket kan illustreras som i Fig. 2.10 a) och till i Fig. 2.11. Denna definition skiljer sig från en annan ganska vanlig definition av sampling som illustreras i Fig. 2.10 b) och Fig. 2.11 där

$$g_D(n) = g(t)|_{t=nT} = g(nT), \quad n = \dots, -1, 0, 1, 2, \dots \quad (2.44)$$



Figur 2.10. a) Modell av sampling. b) Alternativ modell av sampling.

I fortsättningen kallar vi $g_D(n)$ för den **diskreta** representationen av $g(t)$. Den samplade signalen $s(t)$ och den diskreta signalen $g_D(n)$ skiljer sig i två avseenden

- $s(t)$ är definierad utefter hela t -axeln medan $g_D(n)$ bara är definierad i ett antal ekvidistanta punkter, $n = \dots, -2, -1, 0, 1, 2, \dots$
- I dessa punkter gäller att

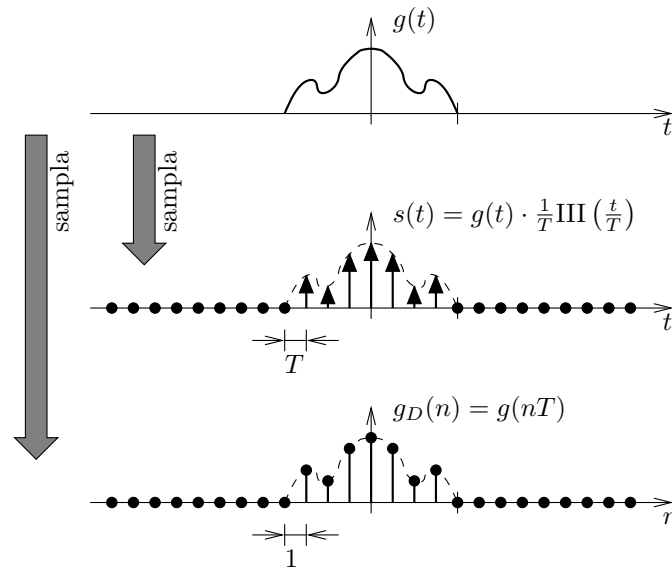
$$s(t) = g_D(n) \cdot \delta(t - nT), \quad t = nT, \quad (2.45)$$

dvs den samplade signalen $s(t)$ är lika med den diskreta signalen $g_D(n)$ multiplicerad med dirac-pulser.

Multiplikationen i signaldomänen motsvaras av faltning i fourierdomänen. Därför kan vi med utnyttjande av impulstågets fouriertransform (2.13) skriva den samplade signalens fouriertransform som

$$S(f) = \text{III}(fT) * G(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} G \left(f - \frac{k}{T} \right), \quad (2.46)$$

vilket illustreras i Fig. 2.12. Utseendet kommer av att faltning med de diskreta dirac-pulserna i impulståget ger upprepade kopieringar av $(1/T)G(f)$.



Figur 2.11. Illustration av de två modellerna att sampla.

Fouriertransformen $S(f)$ i Fig. 2.12 avviker tyvärr avsevärt från $G(f)$ i form av ett överlapp eller **vikningsdistorsion**. Detta är betänkligt eftersom vi i vår digitaliserade värld måste använda diskreta sampel för att representera kontinuerliga signaler. Om vi ökar samplingsfrekvensen $1/T$ som i Fig. 2.13 a)-c) förbättras emellertid situationen väsentligt. I fourierdomänen är nu de olika kopiorna av $G(f)$ inte längre överlappande förutsatt att $g(t)$ är bandbegränsad, dvs saknar frekvenser över en viss högsta frekvens W . Detta ger oss samplingsteoremet.

Samplingsteoremet

Låt signalen $g(t)$ samplas till $s(t)$. Om samplingsfrekvensen $1/T$ är större än 2 gånger $g(t)$:s maximala frekvens W , dvs

$$\frac{1}{T} > 2W, \quad (2.47)$$

så kan $g(t)$ rekonstrueras fullständigt i varje punkt av t -axeln med hjälp av $s(t)$.

Tekniken för sådan **rekonstruktion** illustreras av Fig. 2.13 c)-e). I fourierdomänen multiplicerar vi $S(f)$ med rektangelfunktionen $T\Pi(fT)$ och återfår därmed det rena ursprungliga frekvensinnehållet i den kontinuerliga signalen $g(t)$,

$$G(f) = S(f) \cdot T\Pi(fT). \quad (2.48)$$

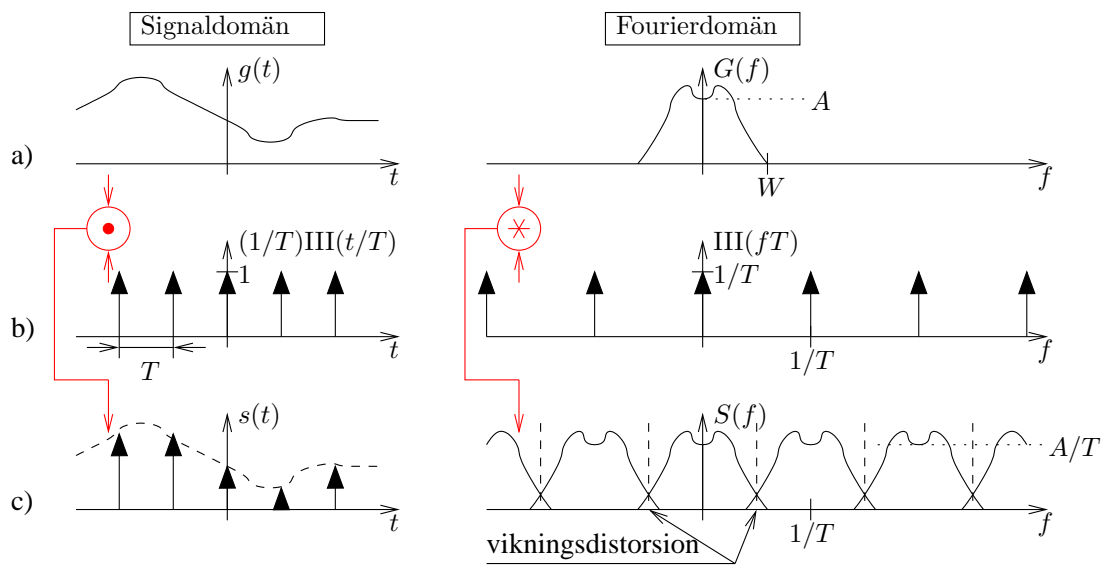
Motsvarande operation i signaldomänen är faltning med rektangelfunktionens (inversa) fouriertransform som är

$$\text{sinc}(t/T) = \frac{\sin(\pi t/T)}{\pi t/T}. \quad (2.49)$$

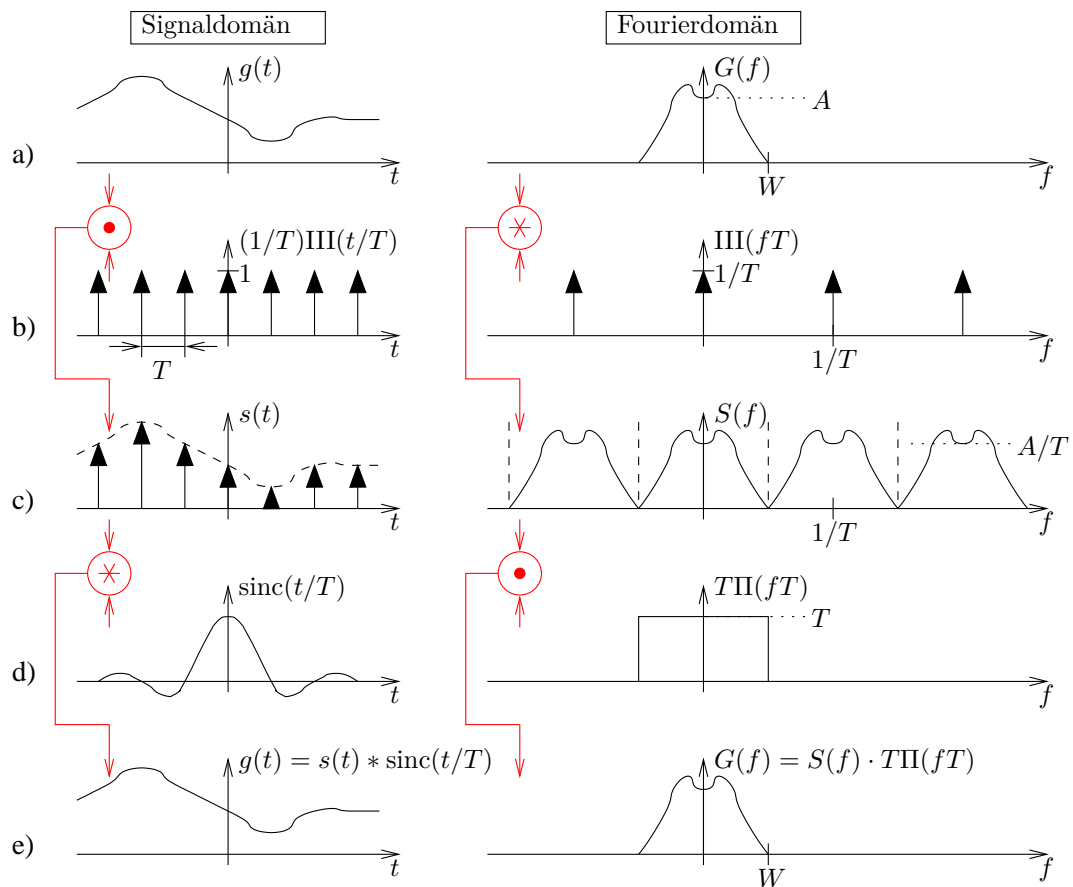
Eftersom sinc-funktionen har oändlig utsträckning är operationen

$$g(t) = s(t) * \text{sinc}(t/T) \quad (2.50)$$

i praktiken omöjlig att utföra exakt. Ekvation (2.49) är dock en viktig teoretisk modell gentemot vilken praktiska, approximativt riktiga, metoder kan utvärderas.



Figur 2.12. Samplingsteoremet illustrerat. Om samplingsfrekvensen $1/T$ är för låg får man vinkningsdistorsion. Korrekt rekonstruktion är då inte möjlig.



Figur 2.13. Samplingsteoremet illustrerat. Om samplingsfrekvensen $1/T$ är tillräckligt hög är perfekt rekonstruktion möjlig.

2.6 Fönstring

En avvikelse från den ideala situation som utmålas i Fig. 2.13 är det faktum att vi inte kan känna till eller mäta upp en signal $g(t)$ eller dess samplade variant från $-\infty$ till $+\infty$. Vi får nöja oss med ett begränsat intervall, ett fönster. I Fig. 2.14 har vi illustrerat detta på så sätt att en signal

$$g(t) = \cos(2\pi f_0 t)$$

har multiplicerats med rektangelfönstret

$$\Pi\left(\frac{t}{NT}\right),$$

där NT är fönstrets utsträckning utefter t -axeln. Den fönsterbegränsade signalen kallar vi $r(t)$ och den fås nu som

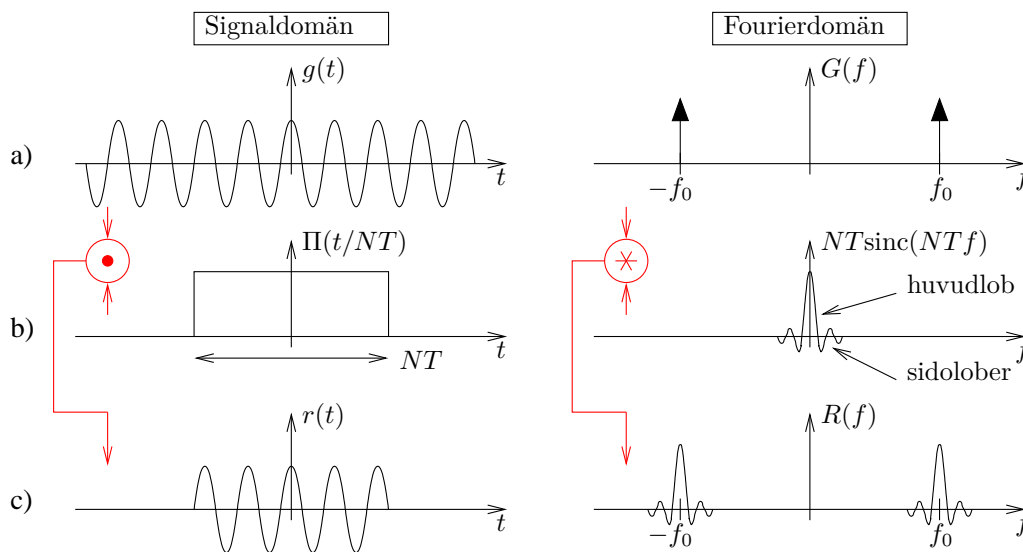
$$r(t) = g(t) \cdot \Pi\left(\frac{t}{NT}\right).$$

I fourierdomänen har vi

$$G(f) = \frac{1}{2}\delta(f + f_0) + \frac{1}{2}\delta(f - f_0),$$

$$NT\text{sinc}(NTf) = \mathcal{F}\left[\Pi\left(\frac{t}{NT}\right)\right], \quad (2.51)$$

$$R(f) = G(f) * NT\text{sinc}(NTf) = \frac{1}{2}\text{sinc}(NT(f + f_0)) + \frac{1}{2}\text{sinc}(NT(f - f_0)).$$



Figur 2.14. Fönstring av en cosinussignal. De distinkta frekvenserna vid $\pm f_0$ breddas. Det är önskvärt med så smal huvudlob och så små sidolober som möjligt.

Skillnaden mellan $G(f)$ och $R(f)$ härrör från sinc-funktionen i (2.51). Med mycket stort fönster NT blir denna sinc-funktion smal som en dirac-puls och $R(f)$ blir då mycket lik $G(f)$, dvs $R(f) \approx G(f)$. Ett litet NT -fönster breddar emellertid dessa funktioner och förvanskar frekvensinnehållet avsevärt, dvs

$$R(f) \neq G(f).$$

Ett annat sätt att se på fönstring är hur det påverkar frekvensinnehållet. De distinkta frekvenserna vid $\pm f_0$ breddas efter fönstring. Som indikerats i figuren har sinc-funktionen $NT\text{sinc}(NTf)$ en huvudlob och flera sidolober. För att bevara de distinkta frekvenserna vid $\pm f_0$ är det önskvärt att huvudloben är så smal som möjligt och att sidoloberna är så små som möjligt. Ibland är det ena önskemålet viktigare än det andra. Det finns flera andra fönster nämnda i litteraturen, t ex hamming-, hanning- och blackman-fönster. Dessa ger mindre sidolober till priset av en breddad huvudlob.

2.7 Tidsdiskret fouriertransform (TDFT)

Funktionen $s(t)$ är en kontinuerlig signal som har värden skilda från 0 endast i sampelpunkterna. Fouriertransformen av $s(t)$ kan räknas ut enligt

$$\begin{aligned} S(f) &= \int_{-\infty}^{\infty} s(t)e^{-j2\pi tf} dt = \int_{-\infty}^{\infty} [\sum_{n=-\infty}^{\infty} g(t)\delta(t-nT)] e^{-j2\pi tf} dt \\ &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} g(t)\delta(t-nT)e^{-j2\pi tf} dt = \sum_{n=-\infty}^{\infty} g(nT)e^{-j2\pi nTf} \end{aligned} \quad (2.52)$$

där vi använt (2.12) och (2.43). Ekvation (2.52) med (2.44) insatt ger

$$S(f) = \sum_{n=-\infty}^{\infty} g_D(n)e^{-j2\pi nTf}. \quad (2.53)$$

Den samplade signalens fouriertransform $S(f)$ kan alltså uttryckas som en summa av viktade diskreta sampelpunkter $g_D(n)$. Genom att göra variabelbytet

$$\Omega = 2\pi fT \quad (2.54)$$

i (2.53) fås en annan vanlig fouriertransform, nämligen den tidsdiskreta fouriertransformen, vars definition är

$$G_T(\Omega) = \sum_{n=-\infty}^{\infty} g_D(n)e^{-j\Omega n}. \quad (2.55)$$

Den inversa tidsdiskreta fouriertransformen ges av

$$g_D(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} G_T(\Omega)e^{j\Omega n} d\Omega \quad (2.56)$$

som lämnas utan bevis. Den tidsdiskreta fouriertransformen $G_T(\Omega)$ säges ha normerad vinkelfrekvens. Dess period är alltid 2π och därför oberoende av sampelavståndet T .

2.8 Diskret fouriertransform (DFT) och cirkulär faltning

Den diskreta fouriertransformen (DFT) kan beräknas som en ändlig summa över en begränsad del av $g_D(n)$,

$$G_D(k) = \sum_{n=0}^{N-1} g_D(n)e^{-j2\pi nk/N}, \quad 0 \leq k \leq N-1. \quad (2.57)$$

Den inversa diskreta fouriertransformen (IDFT) utföres som

$$g_D(n) = \frac{1}{N} \sum_{k=0}^{N-1} G_D(k)e^{j2\pi nk/N}, \quad 0 \leq n \leq N-1. \quad (2.58)$$

som lämnas utan bevis. Faktorn $1/N$ förekommer i IDFT men inte i DFT. För DFT gäller en mängd teorem liknande dem som gäller för kontinuerlig fouriertransform och TDFT. Exempel på sådana teorem är skalningsteoremet och skiftteoremet som finns beskrivna i bl a [6] och för 2D i Tab. 3.2. Något som är specifikt för DFT:n är dess faltningsteorem.

Teorem för cirkulär faltning

Det gäller att multiplikation DFT-domänen motsvaras av cirkulär faltning i den andra domänen,

$$(g_D *_N h_D)(m) = \sum_{n=0}^{N-1} g_D(n)h_D((m-n)_N) \Leftrightarrow G_D(k) \cdot H_D(k), \quad (2.59)$$

där $*_N$ noterar cirkulär faltning och $()_N$ betecknar modulo N operation, dvs $h_D()$ kan uppfattas som periodiskt upprepade eller cirkulär.

Bevis

Antag att vi har två sekvenser $g_D(n)$ och $h_D(n)$ med längden N . Deras respektive DFT:er definieras som

$$G_D(k) = \sum_{n=0}^{N-1} g_D(n) e^{-jk(2\pi/N)n}, \quad k = 0, \dots, N-1, \quad (2.60)$$

$$H_D(k) = \sum_{n=0}^{N-1} h_D(n) e^{-jk(2\pi/N)n}, \quad k = 0, \dots, N-1. \quad (2.61)$$

Om dessa två DFT:er multipliceras ihop fås ett resultat med längden N som vi kan kalla $Y_D(k)$. Låt oss nu bestämma relationen mellan sekvenserna $g_D(n)$ och $h_D(n)$. Vi har

$$Y_D(k) = G_D(k) \cdot H_D(k). \quad (2.62)$$

Invers DFT av detta blir

$$y_D(m) = \frac{1}{N} \sum_{k=0}^{N-1} Y_D(k) e^{jk(2\pi/N)m} = \frac{1}{N} \sum_{k=0}^{N-1} G_D(k) H_D(k) e^{jk(2\pi/N)m}. \quad (2.63)$$

Sätt nu in (2.60) och (2.61) i (2.63). Detta ger

$$y_D(m) = \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum_{n=0}^{N-1} g_D(n) e^{-jk(2\pi/N)n} \right] e^{jk(2\pi/N)m} \quad (2.64)$$

$$y_D(m) = \frac{1}{N} \sum_{n=0}^{N-1} g_D(n) \sum_{l=0}^{N-1} h_D(l) \left[\sum_{k=0}^{N-1} e^{jk(2\pi/N)(m-n-l)} \right]. \quad (2.65)$$

Summan innanför klamrarna i (2.65) kan nu skrivas

$$\sum_{k=0}^{N-1} e^{jk(2\pi/N)(m-n-l)} = \begin{cases} N, & l = m - n - pN = (m - n)_N, \quad p \text{ heltal} \\ \frac{1 - e^{j2\pi(m-n-l)/N}}{1 - e^{j2\pi(m-n-l)/N}} = 0, & \text{f.ö.} \end{cases} \quad (2.66)$$

Om nu (2.66) sätts in i (2.65) fås

$$y_D(m) = (g_D *_N h_D)(m) = \sum_{n=0}^{N-1} g_D(n) h_D[(m - n)_N] \quad (2.67)$$

vilket tillsammans med (2.62) bevisar (2.59).

V.S.V**2.9 Från kontinuerlig till diskret fouriertransform (DFT)**

Fig. 2.15 illustrerar nu hela sambandet mellan den kontinuerliga fouriertransformen och DFT. Vi börjar med den kontinuerliga signalen $g(t)$ och dess fouriertransform $G(f)$. Vi tänker oss att $g(t)$ är begränsad och att $G(f)$ är "tillräckligt begränsad" så att största delen av $G(f)$:s energi ligger i intervallet $|f| < 1/(2T)$. Det går inte att anta att både $g(t)$ och $G(f)$ är begränsade. Man kan visa att om $g(t)$ är begränsad så kan inte $G(f)$ vara begränsad och vice versa, jämför gärna med fourierparen i Fig. 2.3 och Fig. 2.4. Funktionen $G(f)$ kan egentligen ha både en realdel och en imaginärdel (det kan också $g(t)$ ha), men i Fig. 2.15 har vi valt att inte visa detta.

Först samplas $g(t)$ till $s(t)$ genom multiplikation med impulståget vilket ger

$$s(t) = g(t) \cdot \frac{1}{T} \text{III} \left(\frac{t}{T} \right). \quad (2.68)$$

Detta ger en upprepning i fourierdomänen enligt

$$S(f) = G(f) * \text{III}(fT) = \frac{1}{T} \sum_{k=-\infty}^{\infty} G \left(f - \frac{k}{T} \right) \quad (2.69)$$

och vi ser i Fig. 2.15 att

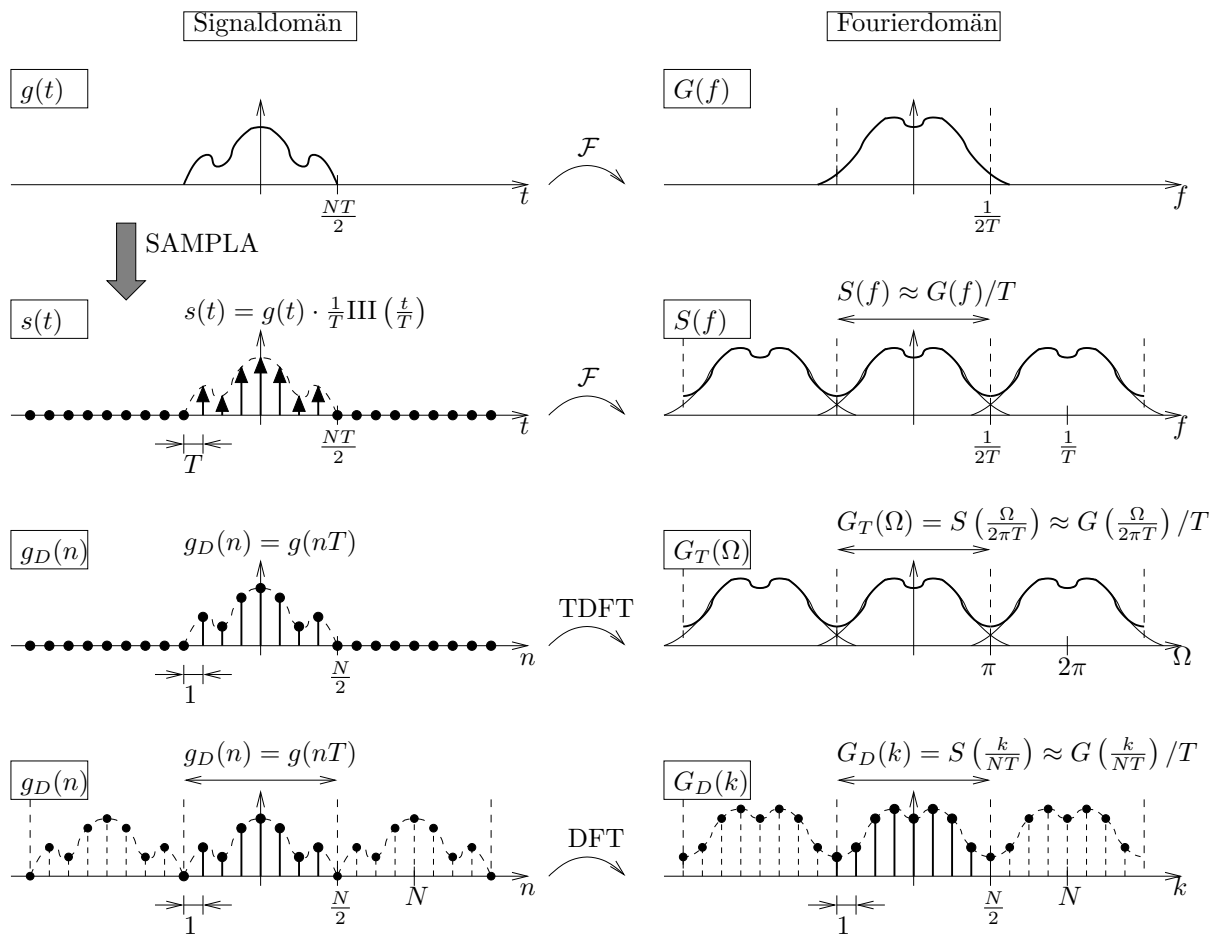
$$S(f) \approx \frac{1}{T} \cdot G(f), \quad -\frac{1}{2T} < f < \frac{1}{2T}. \quad (2.70)$$

Det tidsdiskreta fourierparet $g_D(n)$, $G_T(\Omega)$ från ekvation (2.55) visas också i figuren. Notera att $G_T(\Omega)$ är en frekvensskalad variant av $S(f)$ med skalfaktorn $\Omega = 2\pi fT$. Detta inses genom att kombinera (2.53), (2.54) och (2.55) vilket ger sambandet

$$G_T(\Omega) = S\left(\frac{\Omega}{2\pi T}\right) \quad (2.71)$$

och vi ser i figuren att

$$G_T(\Omega) \approx \frac{1}{T} G\left(\frac{\Omega}{2\pi T}\right). \quad (2.72)$$



Figur 2.15. Samband mellan kontinuerlig fouriertransform, samplad kontinuerlig fouriertransform, TDFT och DFT.

För att den diskreta fouriertransformen (DFT) ska vara symmetrisk runt origo och lättare gå att jämföra med de övriga funktionerna i Fig. 2.15, definierar vi den något annorlunda än tidigare i (2.57) och (2.58). DFT:n definieras nu som

$$G_D(k) = \sum_{n=-N/2}^{N/2-1} g_D(n) e^{-j2\pi nk/N}, \quad -\frac{N}{2} \leq k \leq \frac{N}{2} - 1, \quad (2.73)$$

där k och n är heltal. Den inversa diskreta fouriertransformen får då utseendet

$$g_D(n) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} G_D(k) e^{j2\pi nk/N}, \quad -\frac{N}{2} \leq n \leq \frac{N}{2} - 1, \quad (2.74)$$

som enkelt kan bevisas genom att ersätta $G_D(k)$ med högerledet av (2.73). Formlerna för DFT och IDFT för funktionerna $g_D(n)$ och $G_D(k)$ gäller alltså exakt. Om man kontrollerar dessa funktioners värde utanför definitionsområdena $-N/2 \leq k, n \leq N/2 - 1$ upptäcker man att de är periodiska. Detta beror i sin tur på att $e^{j\phi}$ är en periodisk funktion. Det repetitiva utseendet på funktionerna $g_D(n)$ och $G_D(k)$ är också indikerat i Fig. 2.15. Man kan välja att betrakta funktionerna antingen begränsade till definitionsområdena eller som periodiska. Det senare synsättet är ofta lämpligt för förståelse av uppträdandet hos DFT och IDFT.

Relation mellan kontinuerlig fouriertransform och DFT

Vi vet sedan tidigare att de diskreta sampelvärdena $g_D(n)$ överensstämmer perfekt med den kontinuerliga funktionen $g(t)$ i punkterna $|n| = 0, 1, 2, 3, \dots$ enligt

$$g_D(n) = g(nT), \quad -\frac{N}{2} \leq n \leq \frac{N}{2} - 1. \quad (2.75)$$

På motsvarande sätt finns en perfekt överensstämmelse mellan $G_D(k)$ och $S(f)$ och en ungefärlig överensstämmelse mellan $G_D(k)$ och $G(f)$ enligt

$$G_D(k) = S\left(\frac{k}{NT}\right) \approx \frac{1}{T}G\left(\frac{k}{NT}\right), \quad -\frac{N}{2} \leq k \leq \frac{N}{2} - 1. \quad (2.76)$$

förutsatt att $g_D(n) = 0$ för $n < -N/2$ och $n \geq N/2$. Relationen mellan kontinuerlig frekvens f och diskret frekvens k är alltså

$$f = \frac{k}{NT}, \quad (2.77)$$

där N är antalet sampelpunkter och T är sampelavståndet.

Bevis

Sätt in $f = k/NT$ i (2.70) och (2.53). Dett ger

$$\begin{aligned} \frac{1}{T}G\left(\frac{k}{NT}\right) &\approx S\left(\frac{k}{NT}\right) = \frac{1}{N} \sum_{n=-\infty}^{\infty} g_D(n)e^{-j2\pi nk/N} = \{g_D(n) = 0 \text{ för } n < -N/2 \text{ och } n \geq N/2\} \\ &= \frac{1}{N} \sum_{n=-N/2}^{N/2-1} g_D(n)e^{-j2\pi nk/N} = \{(2.73)\} = G_D(k) \end{aligned} \quad (2.78)$$

V.S.V

Ekvation (2.76) visar att fouriertransformen $G(f)$ kan beräknas approximativt i N punkter genom att först sampla $g(t)$ och därefter applicera en DFT på dessa diskreta sampelpunkter. För att DFT-värdena $G_D(k)$ ska överensstämma med $G(f)$ måste de dock skalas på både höjden och bredden enligt ekvation (2.76). Notera att den kontinuerliga fouriertransformen $G(f)$ aldrig kan återskapas perfekt ur den diskreta fouriertransformen $G_D(k)$ eftersom sambandet mellan $G_D()$ och $G()$ är approximativt. Någon kanske argumenterar att detta skulle gå om $G(f)$ vore bandbegränsad. I ett sådant fall är dock $g(t)$ inte begränsad. Eftersom N är ett ändligt tal kommer sampelpunkterna i $g_D(n)$, $-N/2 \leq n \leq N/2 - 1$, inte att räcka för hela den oändligt långa $g(t)$. Den obegränsade funktionen $g(t)$ måste därför först begränsas genom fönstring till $g_{\text{fönstr}}(t)$. Fouriertransformen av denna funktion, $G_{\text{fönstr}}(f)$, blir då obegränsad och vi är tillbaka till fallet som beskrivs i Fig. 2.15.

2.10 Diskret faltning och faltningskärnor

Faltning av två kontinuerliga signaler definieras som

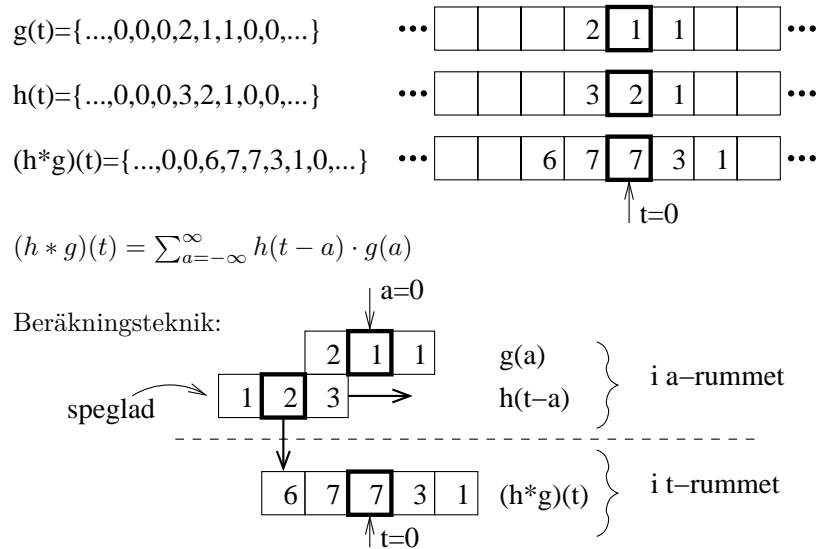
$$y(t) = (h * g)(t) = \int_{-\infty}^{\infty} h(t-a)g(a) da. \quad (2.79)$$

Om nu h och g är diskreta signaler, dvs sekvenser av sampel förvandlas integralen till en summa med uttrycket

$$y(t) = \sum_{a=-\infty}^{\infty} h(t-a)g(a) = \sum_{a=-\infty}^{\infty} h(a)g(t-a). \quad (2.80)$$

Notera att vi, trots att det rör sig om diskreta signaler, har valt att kalla variabeln t istället för som tidigare n . Minst en av de två ingångssekvenserna $h(t)$, $g(t)$ brukar vara ändlig, dvs begränsad till N st värden skilda från 0.

I Fig. 2.16 visas ett fall med numeriska värden där alla tre sekvenserna är ändliga, och därmed 0 före och efter de ändliga intervallen. Faltningformeln ger att beräkningen sker i a -rummet där h -sekvensen speglas och glider framåt för ökande värden på t .



Figur 2.16. Diskret faltning.

Faltning sker oftast som en operation av en relativt kort sekvens (sekvensen h) på en längre, potentiellt oändlig sekvens (sekvensen g) i Fig. 2.16. I ett sådant fall kallas sekvensen h **faltningskärna**, (engelska: convolution kernel), **faltningsoperator**, **linjär operator** eller kort och gott **filter**.

Det kan nu vara lämpligt att något diskutera olika en-dimensionella faltningkärnors filtreringsegenskaper. Mera om detta kommer att sägas längre fram då vi behandlar tvådimensionella signaler. Vi använder beteckningen

$$\{a, b\}/k \text{ istället för } \{a/k, b/k\}$$

Storheten nedanför snedstrecket är sålunda en (normaliserings-)faktor som är gemensam för alla koefficienterna i faltningkärnan. Operatorerna

$$\{1, 1\}/2 \quad \{1, 2, 1\}/4 \quad \{1, 3, 3, 1\}/8 \quad \{1, 4, 6, 4, 1\}/16$$

är alla medelvärdesbildande i viss mening. De bevarar DC-komponenten (medelvärdet) i signalen medan högfrekventa komponenter undertryckes. Eftersom de alla fyra är jämna funktioner (med origo symmetriskt placerat) blir deras fouriertransformer reella och jämna funktioner, se Fig. 2.17. Notera dock att för $\{1, 1\}/2$ blir ut-resultatet förskjutet ett halvt sampelavstånd jämfört med in-sampelvärdena. Detta är något som programmeraren måste ha kontroll på.

Hur gör man då för att beräkna en faltningkärnas fouriertransform? Nedan härleds den kontinuerliga fouriertransformen, TDFT:n och DFT:n för faltningkärnan $\{1, 2, 1\}/4$. Vi förutsätter att origo ligger i mitten av kärnan så att

$$h_D(n) = \begin{cases} 1/4, 1/2, 1/4, & n = -1, 0, 1 \\ 0, & \text{för övrigt.} \end{cases}$$

Kontinuerlig fouriertransform av faltningskärnan $\{1, 2, 1\}/4$

- Sätt dirac-pulser på varje element i operatoren och antag sampelavstånd T . Detta ger den kontinuerliga funktionen

$$h(t) = [\delta(t + T) + 2\delta(t) + \delta(t - T)]/4.$$

- Fouriertransformera $h(t)$ till

$$H(f) = \frac{1}{4}e^{j2\pi Tf} + \frac{1}{2} + \frac{1}{4}e^{-j2\pi Tf} = \cos(2\pi Tf)/2 + 1/2 = \cos^2(\pi Tf).$$

- Operatoren $\{1, 2, 1\}/4$ har alltså fouriertransformen $\cos^2(\pi Tf)$.

TDFT av faltningskärnan $\{1, 2, 1\}/4$

- Sätt in faltningskärnans värden $h_D(n)$ i TDFT-formeln (2.55)

$$H_T(\Omega) = \sum_{n=-\infty}^{\infty} h_D(n)e^{-j\Omega n} = \frac{1}{4}e^{-j\Omega(-1)} + \frac{1}{2}e^{-j\Omega 0} + \frac{1}{4}e^{-j\Omega 1} = \cos^2\left(\frac{\Omega}{2}\right)$$

- Operatoren $\{1, 2, 1\}/4$ har alltså TDFT:n $\cos^2(\Omega/2)$.

DFT av faltningskärnan $\{1, 2, 1\}/4$

- Sätt in faltningskärnans värden $h_D(n)$ i DFT-formeln (2.73)

$$H_D(k) = \sum_{n=-N/2}^{N/2-1} h_D(n)e^{-j2\pi k \frac{n}{N}} = \frac{1}{4}e^{-j2\pi k \frac{-1}{N}} + \frac{1}{2}e^{-j2\pi k \frac{0}{N}} + \frac{1}{4}e^{-j2\pi k \frac{1}{N}} = \cos^2\left(\frac{\pi k}{N}\right)$$

- Operatoren $\{1, 2, 1\}/4$ har alltså DFT:n $\cos^2(\pi k/N)$. Notera att N är en fri parameter.

Alla tre transformerna ger likvärdiga resultat, som dock är skalade i förhållande till varandra. I Fig. 2.17 föredrog vi dock att använda den kontinuerliga fouriertransformen av faltningskärnan eftersom den är lättare att jämföra med kontinuerlig fouriertransform av kontinuerliga funktioner.

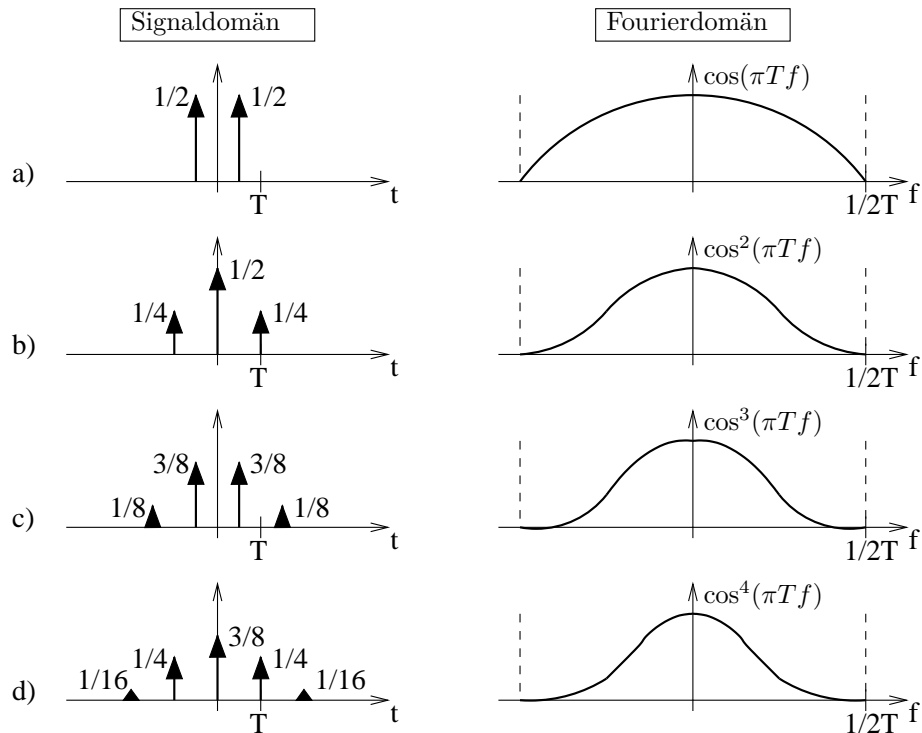
Differentieringsoperatoren $\{1, -1\}/T$ har ingen DC-komponent. Däremot förstärker den (ger utslag för, detekterar) förekomster av ojämnheter, språng och partier av stark lutning i den signal som den appliceras på. En liknande effekt har operatoren $\{1, 0, -1\}/2T$ fast med mindre känslighet för små ojämnheter. Att denna operator verkligen beräknar den approximativa derivatan illustreras i Fig. 2.18. Till vänster visas att derivatan $g'(t_1)$ approximativt beräknas som

$$g'(t_1) \approx \frac{g(t_1 + T) - g(t_1 - T)}{2T}.$$

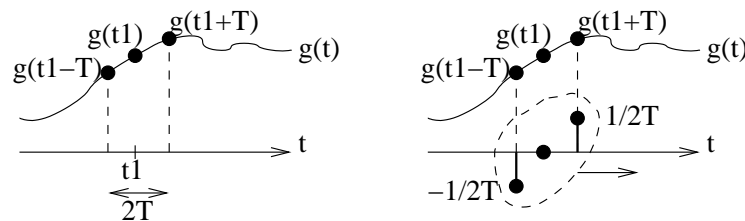
Till höger visas att faltning med $\{1, 0, -1\}/2T$ ger precis samma resultat, dvs

$$-\frac{1}{2T} \cdot g(t_1 - T) + \frac{1}{2T} \cdot g(t_1 + T) = \frac{g(t_1 + T) - g(t_1 - T)}{2T}.$$

En nackdel med $\{1, -1\}/T$ är att utgångsresultatet blir förskjutet ett halvt sampelavstånd jämfört med ingångsresultatet.



Figur 2.17. Medelvärdesbildande faltningsskärnor och deras fouriertransformer.



Figur 2.18. Faltning med $\{1, 0, -1\}/2T$ motsvarar approximativ derivering.

Operatorerna och deras fouriertransformer kan studeras i Fig. 2.19. Följande regel gäller som bekant för fouriertransformen vid derivering av en funktion,

$$\mathcal{F}\left[\frac{d}{dt}[g(t)]\right] = \mathcal{F}[g'(t)] = j2\pi f \cdot G(f). \quad (2.81)$$

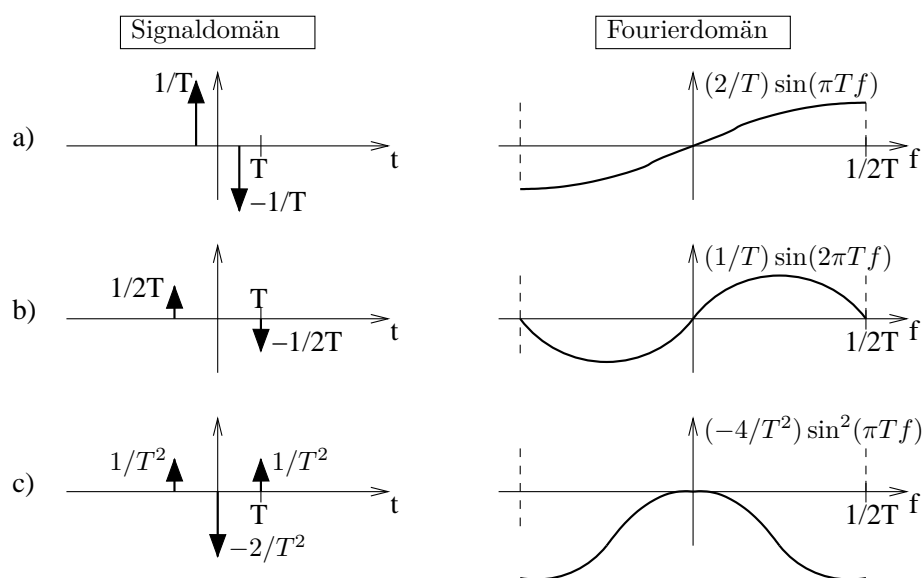
Eftersom derivering gör att fouriertransformen multipliceras med $j2\pi f$ kan man se det som om deriveringsoperationen utför en faltning. Derivering kan därmed skrivas som en faltning enligt

$$\frac{d}{dt}[g(t)] = \frac{d}{dt} * g(t). \quad (2.82)$$

Deriveringsoperatoren d/dt har alltså fouriertransformen $j2\pi f$. Differentieringsoperatorerna i Fig. 2.19 följer denna linjära funktion i närheten av origo. Dubbel differentiering får vi om vi t ex använder operatoren $\{1 -1\}/T$ två gånger vilket är liktydigt med att applicera operatoren

$$\{1, -2, 1\}/T^2 = \{1, -1\}/T * \{1, -1\}/T.$$

Denna operator visas med sin fouriertransform längst ner i Fig. 2.19. I närheten av origo följer den 2:a derivatorns fouriertransform $(j2\pi f)^2 = -4\pi^2 f^2$ som sig bör. Vi ska upprepa och utvidga diskussionen om derivatorer i avsnitt 3.9.



Figur 2.19. Differentierande faltningskärnor med fouriertransformer.

Kapitel 3

Tvådimensionell signalbehandling

I detta kapitel och de följande kommer vi in på två-dimensionell signalbehandling, dvs bildbehandling eller engelskans "image processing". Det finns mycket litteratur inom området, både tidskriftsartiklar och böcker och vi väljer att som exempel referera till några av böckerna [3], [4], [5], [7], [8] och [11].

3.1 2D fouriertransform

En endimensionell signal är en funktion som endast beror av en variabel. Ett exempel på detta är sinusvågen,

$$x(t) = \sin(t). \quad (3.1)$$

På liknande sätt är en n-dimensionell signal en funktion som beror av n st oberoende variabler. Ett exempel på en två-dimensionell signal är

$$f(x, y) = \sin(x - y) + 1. \quad (3.2)$$

Inom signalbehandlingen begränsar man sig ofta till en-dimensionella signaler med tiden t som variabel. Inom bildbehandlingen arbetar man huvudsakligen med två-dimensionella signaler. Det vi i dagligt tal menar med en bild är ju en två-dimensionell funktion. De oberoende variablerna betecknas ofta med x och y och betraktas som spatiala (rums-) koordinater. Den två-dimensionella signalen $f(x, y)$ i (3.2) illustrerades i Fig. 1.1, där $f(x, y) = 0$ avbildas på svart, $f(x, y) = 2$ avbildas på vitt och övriga värden på $f(x, y)$ med proportionella gråskalevärden.

Generaliseringen av från en-dimensionell till två-dimensionell signalteori är i långa stycken rättfram och relativt självklar. Oavsett detta bör utvidgningen till två dimensioner göras och noteras. Till detta kommer att effekter och samband existerar för tvådimensionella signaler som helt saknar motsvarighet i en-dimensionell signalbehandling. Ett sådant fall är projektionsteoremet, vilket dock inte behandlas här.

För en kontinuerlig två-dimensionell signal $f(x, y)$ definieras dess fouriertransform $F(u, v)$ med

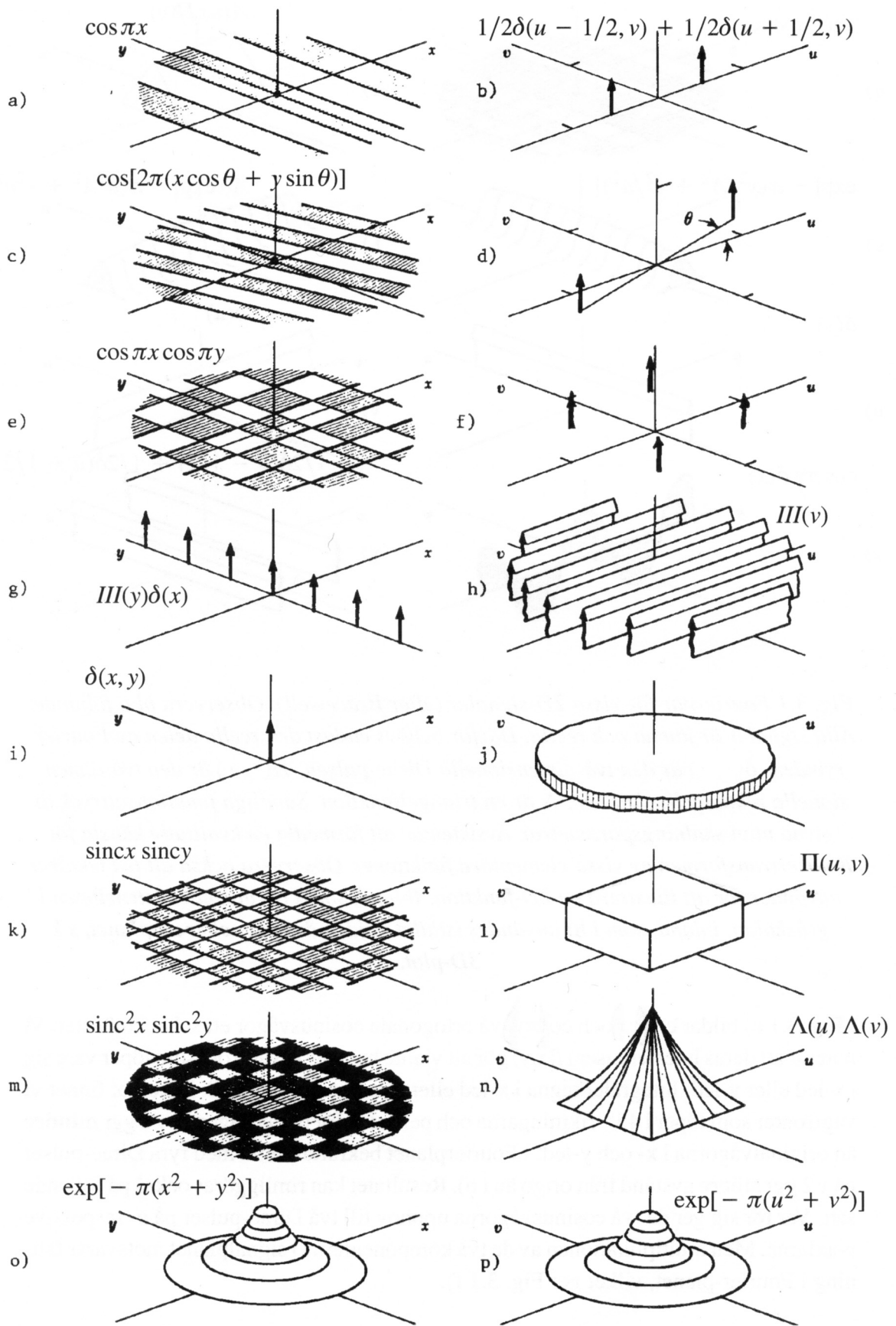
$$\begin{aligned} \mathcal{F}_2[f(x, y)] = F(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(xu+yv)} dx dy \\ &= \int_{-\infty}^{\infty} e^{-j2\pi yv} \left[\int_{-\infty}^{\infty} f(x, y) e^{-j2\pi x u} dx \right] dy \\ &= \int_{-\infty}^{\infty} e^{-j2\pi x u} \left[\int_{-\infty}^{\infty} f(x, y) e^{-j2\pi y v} dy \right] dx. \end{aligned} \quad (3.3)$$

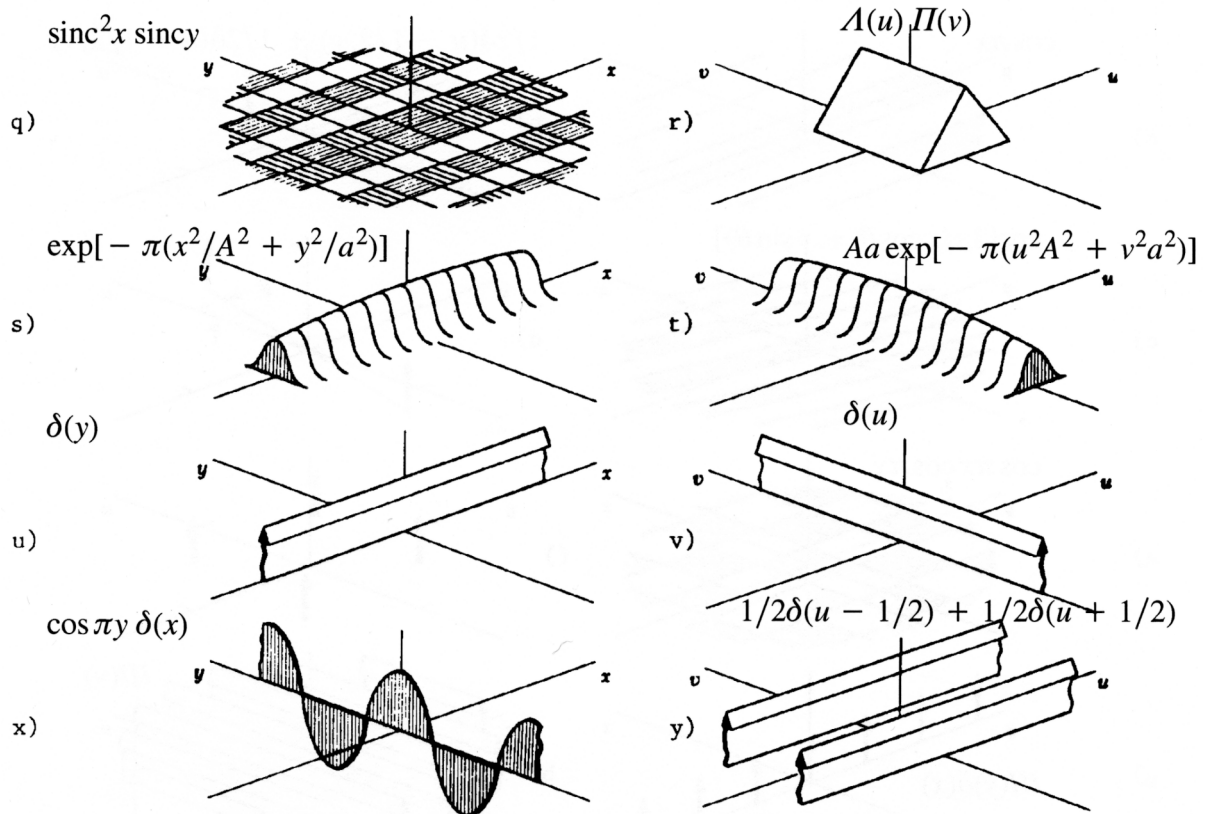
Som framgår av (3.3) är transformen **separerbar**. Vi kan t ex först beräkna 1D-transformen i x-led för varje rad i bilden. På detta resultat appliceras sedan 1D-transformen i y-led för varje kolumn. Detta kommer att illustreras närmare nedan i anslutning till Fig. 3.6.

Den inversa fouriertransformen definieras av

$$\mathcal{F}_2^{-1}[F(u, v)] = f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(xu+yv)} dudv, \quad (3.4)$$

vilket står i full överensstämmelse med det en-dimensionella fallet.





Figur 3.1. Fourierpar för vissa 2D-signaler (efter Bracewell). Observera bl a följande. Alla signaler är jämna och reella. Därför behövs endast den reella delen av fourierrymden. $\delta(\cdot, \cdot)$ är den två-dimensionella Dirac-pulsen, $\Pi(\cdot, \cdot)$ är den två-dimensionella rektangelfunktionen, Λ är en triangelfunktion. Samtliga funktionsuttryck är givna utan skalningsparametrar. Avsikten är att förmedla en kvalitativ känsla för fouriertransformen av vissa elementära funktioner. Observera också att två tekniker användes för att illustrera en 2D-funktion. Ibland, som i a) användes intensiteten i gråskalan, ibland, som i h) användes istället höjden över (u, v) - eller (x, y) -planet i en s k 3D-plot.

Fig. 3.1 visar ett antal två-dimensionella funktioner och deras fouriertransformer. Samtliga är reella, jämna funktioner varav följer att också $F(u, v)$ är jämn och reell. Det kan inte nog understrykas att för att grafiskt beskriva en allmän två-dimensionell fouriertransform krävs två plan för $F_{\text{Re}}(u, v)$ respektive $F_{\text{Im}}(u, v)$. Observera att (u, v) -planet inte är ett komplext talplan.

Exemplen i Fig. 3.1 kan föranleda en hel del tänkvärda observationer. Vi noterar att i Fig. 3.1.a) så finns ingen variation alls i y -led, endast en DC-komponent. Mycket riktigt är i Fig. 3.1.b) också $F(u, v) = 0$ överallt utom på u -axeln där $v = 0$.

De två Dirac-pulserna i Fig. 3.1 b) följer rotationen av cosinusmönstret i Fig. 3.1 c) så att vi i Fig. 3.1 d) finner Dirac-pulserna utefter en linje med vinkeln θ ; samma linje i (x, y) -planet (ej utritad) går vinkelrätt genom vågmönstret $f(x, y)$. I Fig. 3.1 c) är också frekvensen dubblerad.

I Fig. 3.1 e) bildar $\cos(\pi x)$ och $\cos(\pi y)$ två ortogonala cosinusvågor ett schackmönster. Vi noterar att deras interferenser i $f(x, y)$ gör att vi inte har någon vågfront som löper vare sig i x -led eller y -led. Integraler tagna i x -led eller y -led ger resultatet 0. Däremot finner vi vågfronter som löper i 45° -riktningarna och perioden för dessa vågor är $\sqrt{2}$ ggr mindre än originalvågorna i x - och y -led. I fourierplanet bekräftas detta med fyra Dirac-pulser på $\sqrt{2}$ ggr större avstånd från origo än i b). Resultatet kan rimliggöras också på följande sätt. Var för sig ger de två cosinusvågorna upphov till två Dirac-pulser på u - respektive v -axlarna. Men multiplikationen av de två komponenterna i signalplanet motsvarar faltning i fourierplanet, vilket ger Fig. 3.1 f).

För den en-dimensionella **dirac-pulsen** $\delta(x)$ i signalplanet gäller som bekant att

$$\mathcal{F}_1[\delta(x)] = 1,$$

dvs fouriertransformen är konstant utefter hela u -axeln. I Fig. 3.1 i)-j) illustreras detta faktum i det två-dimensionella fallet, dvs

$$\mathcal{F}_2[\delta(x, y)] = \mathcal{F}_2[\delta(x) \cdot \delta(y)] = 1. \quad (3.5)$$

Övriga exempel lämnas till läsarens egen begründan.

Samband för 2D-kontinuerlig och 2D-diskret fouriertransform visas i Tab. 3.1 och Tab. 3.2. Sambanden är ganska rättframma generaliseringar av det endimensionella fallet i föregående avsnitt. I sambanden i Tab. 3.1 nedan noterar \mathbf{A} en godtycklig 2-dimensionell matris och \mathbf{R} en rotationsmatris,

$$\mathbf{R} = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix}, \quad (3.6)$$

där β betecknar rotationsvinkeln.

Teorem	Spatialdomän $x, y \in \mathbf{R}$	Fourierdomän $x, y \in \mathbf{R}$
Definition:	$f(x, y) = \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(xu+yv)} dudv$	$F(u, v) = \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(xu+yv)} dx dy$
Reell Signal:	$f(x, y)$ real	$F(-u, -v) = F^*(u, v)$
Linjäritet:	$af_1(x, y) + bf_2(x, y)$	$aF_1(u, v) + bF_2(u, v)$
Translation, tid:	$f(x-a, y-b)$	$e^{-j2\pi(au+bv)} F(u, v)$
Translation, frekv:	$e^{j2\pi(ax+by)} f(x, y)$	$F(u-a, v-b)$
Skalning:	$f(ax, by)$	$(1/ ab) \cdot F(u/a, v/b)$
Faltning:	$(f * g)(x, y)$	$F(u, v) \cdot G(u, v)$
Korrelation:	$(f \square g)(x, y)$	$F^*(u, v) \cdot G(u, v)$
Multiplikation:	$f(x, y) \cdot g(x, y)$	$(F * G)(u, v)$
Derivata i x:	$\frac{\partial}{\partial x} f(x, y)$	$j2\pi u \cdot F(u, v)$
Derivata i y:	$\frac{\partial}{\partial y} f(x, y)$	$j2\pi v \cdot F(u, v)$
Laplace:	$\nabla^2 f(x, y) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) f(x, y)$	$-4\pi^2(u^2 + v^2) \cdot F(u, v)$
Generell skalning:	$f(\mathbf{Ax}), \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$	$\frac{1}{ \det \mathbf{A} } F((\mathbf{A}^{-1})^T \mathbf{u}), \mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$
Rotation 1:	$f(\mathbf{Rx}), \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$	$F(\mathbf{Ru}), \mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$
Rotation 2:	$f(r, \theta + \theta_0)$ $x = r \cos \theta, y = r \sin \theta$	$F(\omega, \varphi + \theta_0)$ $u = \omega \cos \varphi, v = \omega \sin \varphi$
Parsevals formel:	$\int \int_{-\infty}^{\infty} f(x, y) g(x, y) dx dy =$	$= \int \int_{-\infty}^{\infty} F(u, v) G^*(u, v) dudv$

Tabell 3.1. Teorem och samband för 2-D kontinuerlig fouriertransform.

Liksom i det endimensionella fallet gäller att fouriertransformen av en reell signal är hermitisk. Eftersom en vanlig bild $f(x, y)$ är reell gäller för dess fouriertransform

$$F(u, v) = F^*(-u, -v) \Rightarrow |F(u, v)| = |F(-u, -v)|, \quad (3.7)$$

vilket gör att diagonalt liggande punkter på lika avstånd från origo är lika. Detta illustreras i Fig. 3.2 som visar en bild och dess fouriertransform, $F(u, v) = \text{Re}F(u, v) + j\text{Im}F(u, v)$, amplitudspektrum $|F(u, v)|$ samt fasspektrum $\arg F(u, v)$. Notera att realdelen och amplitudspektrum är jämna, medan imaginärdelen och fasspektrum är udda. Notera också att de låga frekvenskomponenterna dominerar, dvs de har större belopp än de höga frekvenskomponenterna. Detta gäller normalt sett för vanliga bilder.

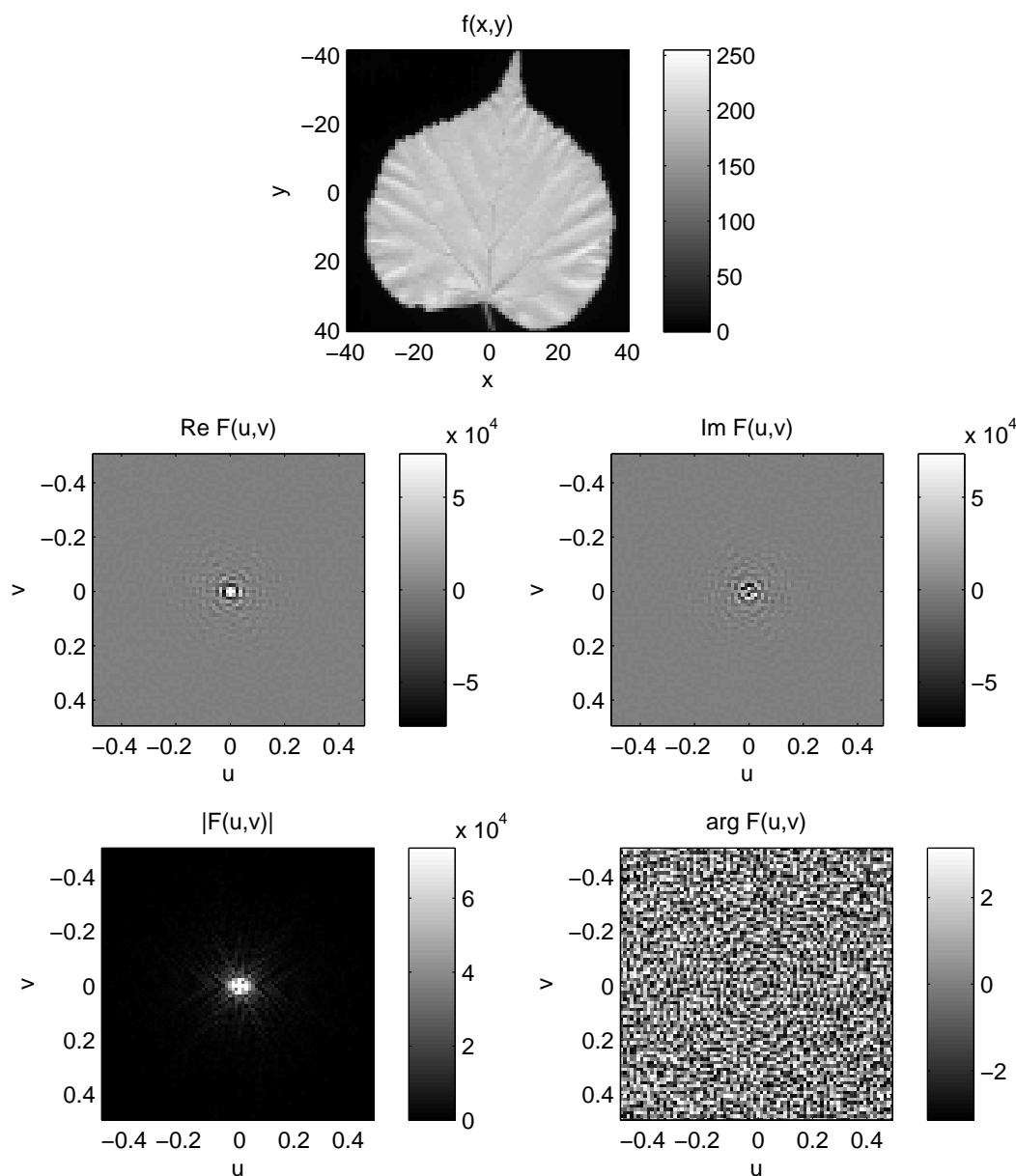
Teorem	Spatialdomän	Fourierdomän
Notation:	$f(x, y)$	$F(u, v)$
Skalningsteomet:	$f(ax, by)$	$(1/ ab) \cdot F(u/a, v/b)$
Cirkulär faltning:	$(f *_{N} g)(x, y)$	$F(u, v) \cdot G(u, v)$
Cirkulär korrelation:	$(f \square_{N} g)(x, y)$	$F^*(u, v) \cdot G(u, v)$
Translationsteomet:	$f(x-a, y-b)$	$e^{-j\frac{2\pi}{N}(au+bv)} F(u, v)$
Parsevals formel:	$\sum \sum_{x,y=0}^{N-1} f(x, y) g(x, y) =$	$= \frac{1}{N^2} \sum \sum_{u,v=0}^{N-1} F(u, v) G^*(u, v)$

Tabell 3.2. Teorem och samband för 2-D diskret fouriertransform.

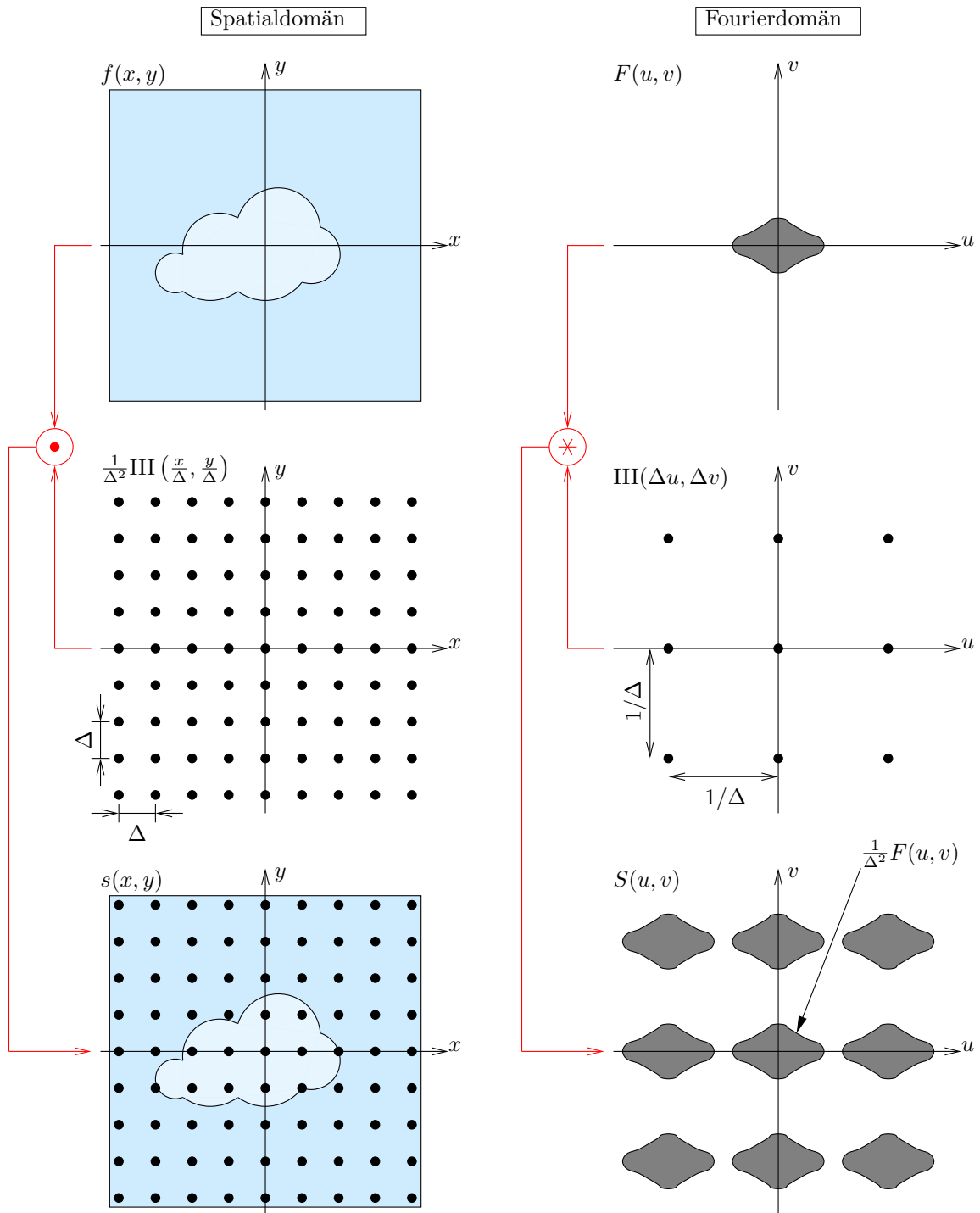
3.2 2D sampling

Sampling av en kontinuerlig signal $f(x, y)$ sker matematiskt med en två-dimensionell motsvarighet till impulståget i (2.12) och avsnitt 2.5, vilken då får utseendet av en "spikmatta", $(1/\Delta^2)\text{III}(x/\Delta, y/\Delta)$, se Fig. 3.3 där spikmattan syns ovanifrån. Samplingstätheterna (Δ_x, Δ_y) behöver inte nödvändigtvis vara lika i x- och y-led, vilket vi dock fortsättningsvis förutsätter, dvs (Δ, Δ) .

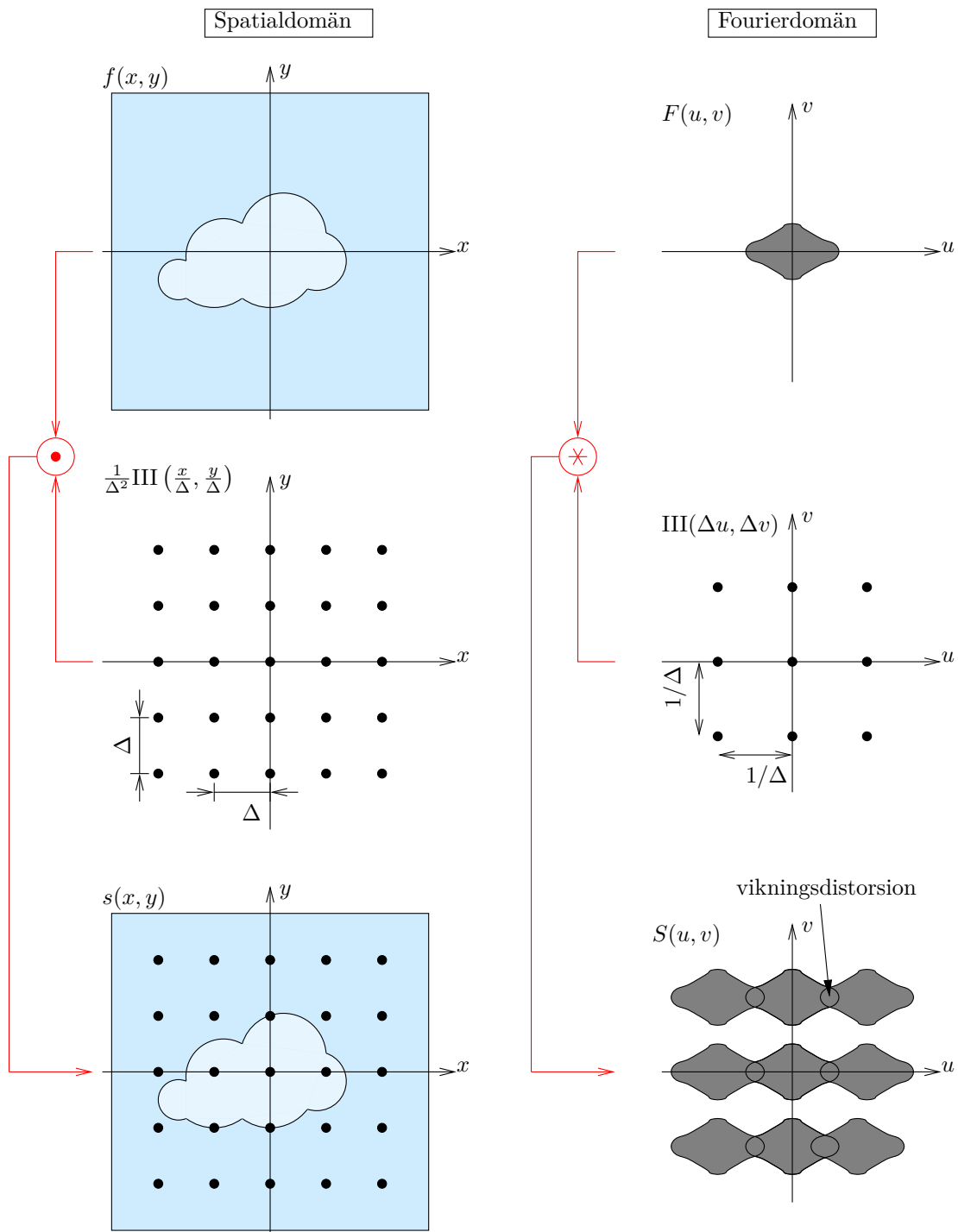
På samma sätt som för en-dimensionella signaler kan vi med tillräckligt hög samplingstäthet representera en **bandbegränsad** kontinuerlig signal utan informationsförlust. **Samplingsteoremet** i två dimensioner illustreras av Fig. 3.3 och Fig. 3.4. Endast om de olika viktade kopiorna av $F(u, v)$ inte överlappar varandra i den samplade funktionens fourierrymd kan det vara möjligt att felfritt rekonstruera $f(x, y)$ i varje punkt i (x, y) -planet. I Fig. 3.3 är samplingsteoremet uppfyllt. Om samlingsavståndet Δ är för stort blir det överlapp, vinkningsdistorsion, vilket illustreras i Fig. 3.4. Felfri rekonstruktion av $f(x, y)$ är då inte möjlig.



Figur 3.2. En bild $f(x, y)$ och dess fouriertransform $F(u, v) = \text{Re}F(u, v) + j\text{Im}F(u, v)$, amplitudspektrum $|F(u, v)|$ samt fasspektrum $\arg F(u, v)$. (Originalbild från lövdata-basen på Avdelningen för Datorseende, Institutionen för Systemteknik, Linköpings Universitet.)



Figur 3.3. Två-dimensionell sampling av $f(x, y)$ med $(1/\Delta^2)\text{III}(x/\Delta, y/\Delta)$ producerar kopior av $F(u, v)$ viktade med $1/\Delta^2$ i fourierdomänen.



Figur 3.4. Två-dimensionell sampling av $f(x, y)$ med $(1/\Delta^2)\text{III}(x/\Delta, y/\Delta)$ producerar kopior av $F(u, v)$ viktade med $1/\Delta^2$ i fourierdomänen. Här är samlingsavståndet Δ för stort, vilket resulterar i vinkningsdistorsion.

Fig. 3.5 illustrerar hur vinkningsdistorsion yttrar sig på en bild och dess fouriertransform. I a) är samplingsfrekvensen tillräcklig, både bild och fouriertransform ser normala ut. I b) har samplingsfrekvensen halverats jämfört med i a). Detta ger vinkningsdistorsion som syns tydligt i de randiga klädesplaggen. I byxorna ser ränderna ut att ha ändrat både bredd och riktning. Man kan även ana vinkningsdistorsionen i fourierdomänen som en ökad intensitet på grund av överlapp i de högre (yttre) frekvenserna.



Figur 3.5. Bilder med fouriertransform. a) Tillräcklig samplingsfrekvens. b) För låg samplingsfrekvensen ger vinkningsdistorsion som syns tydligt i bl a byxornas randning. Vinkningsdistorsionen syns även i fourierdomänen som en ökad intensitet för de högre frekvenserna. (Originalbilden finns på <http://www.imageprocessingplace.com/>. Ursprungligen från Allen Gersho's lab, University of California, Santa Barbara.)

3.3 2D DFT

Två-dimensionell DFT definieras med uttrycket

$$F_D(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f_D(n, m) e^{-j2\pi(nk/N + ml/M)}, \quad 0 \leq k \leq N-1, \quad 0 \leq l \leq M-1, \quad (3.8)$$

Den inversa motsvarigheten IDFT blir

$$f_D(n, m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} F_D(k, l) e^{j2\pi(nk/N + ml/M)}, \quad 0 \leq n \leq N-1, \quad 0 \leq m \leq M-1. \quad (3.9)$$

Symmetrisk DFT är ofta att föredra i bildbehandlingsammanhang, då man vill att origo ska befinna sig mitt i bilden och mitt i fouriertransformen. Två-dimensionell symmetrisk DFT definieras som

$$F_D(k, l) = \sum_{n=-N/2}^{N/2-1} \sum_{m=-M/2}^{M/2-1} f_D(n, m) e^{-j2\pi(nk/N + ml/M)}, \quad -\frac{N}{2} \leq k \leq \frac{N}{2}-1, \quad -\frac{M}{2} \leq l \leq \frac{M}{2}-1, \quad (3.10)$$

där k, l och n, m är heltal. Den två-dimensionella symmetriska inversa diskreta fouriertransformen (2D IDFT) får då utseendet

$$f_D(n, m) = \frac{1}{NM} \sum_{k=-N/2}^{N/2-1} \sum_{l=-M/2}^{M/2-1} F_D(k, l) e^{j2\pi(nk/N + ml/M)}, \quad -\frac{N}{2} \leq n \leq \frac{N}{2}-1, \quad -\frac{M}{2} \leq m \leq \frac{M}{2}-1. \quad (3.11)$$

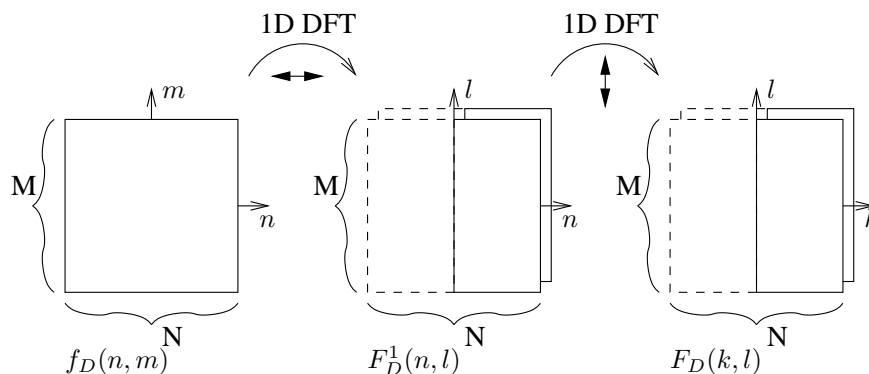
Separering av de två dimensionerna gör att operationen (3.10) kan beräknas i två steg såsom indikeras i Fig. 3.6. Först utföres N st M -punkters DFT på de N kolumnerna i $f_D(n, m)$ vilket ger

$$F_D^1(n, l) = \sum_{m=-N/2}^{N/2-1} f_D(n, m) e^{-j2\pi ml/M}.$$

Därefter utföres M st N -punkters DFT på de M raderna i $F_D^1(n, l)$ vilket ger

$$F_D(k, l) = \sum_{n=-N/2}^{N/2-1} F_D^1(n, l) e^{-j2\pi nk/N},$$

Notera att DFT-data är komplext. Den ena del-bilden är real-delen och den andra är imaginärdelen. Dessa är symmetriska, realdelen är jämn och imaginärdelen är udda. Därmed behöver den streckade delen egentligen inte räknas ut.



Figur 3.6. Tvådimensionell DFT utföres med N st endimensionella transformeringar i m -led följt av M st endimensionella transformeringar i n -led.

3.4 Samband: samplad kontinuerlig och samplad diskret bild

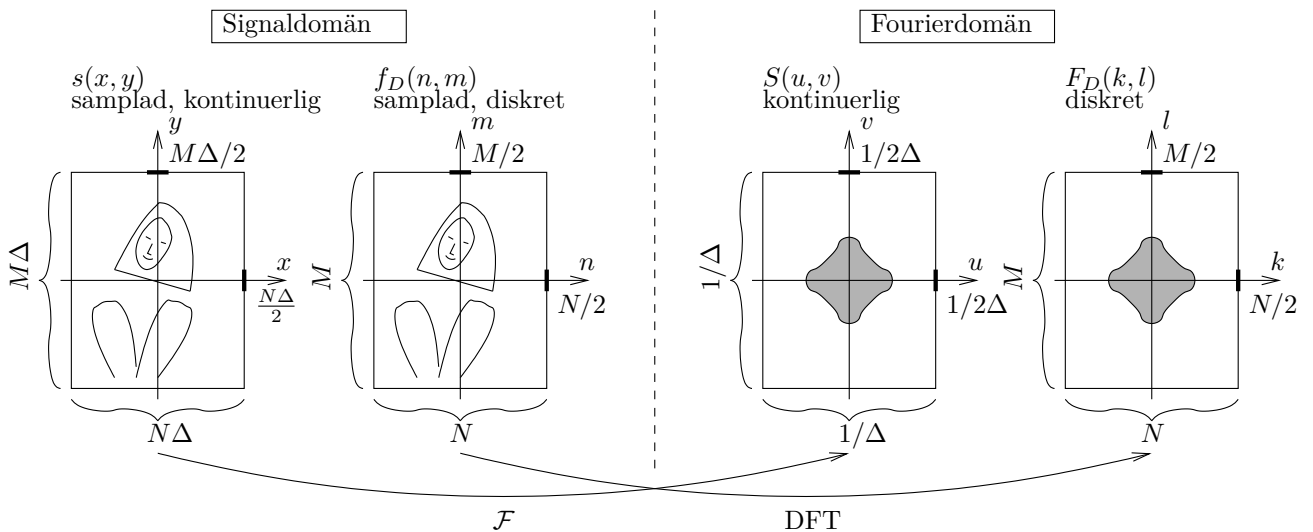
För att få kontroll på storlekar och frekvenser i Fig. 3.5, visar vi Fig. 3.7. Jämför denna figur med det endimensionella fallet i Fig. 2.15, $s(x, y)$ motsvarar $s(t)$, båda funktioner samplade med impulståget, och $f_D(n, m)$ motsvarar $g_D(n)$, båda diskreta samplade funktioner. Vi antar att den symmetriska DFT:n enligt (3.10) har använts så att origo ligger i centrum av bild och fouriertransform. Det gäller att maximal spatiell frekvens $u = 1/(2\Delta)$ ($v = 1/(2\Delta)$) motsvarar $k = N/2$ ($l = M/2$), dvs

$$\begin{cases} u = 1/(2\Delta) \iff k = N/2, \\ v = 1/(2\Delta) \iff l = M/2, \end{cases} \quad (3.12)$$

eller annorlunda uttryckt: För en diskret fouriertransform gäller att positionen k (l) motsvaras av den bakomliggande spatiella frekvensen

$$\begin{cases} u = k/(N\Delta), \\ v = l/(M\Delta). \end{cases} \quad (3.13)$$

Hur yttrar sig då detta på bilderna i Fig. 3.5? Där är antalet sampelpunkter hälften så stort i b) som i a), $(N_b, M_b) = (0.5 \cdot N_a, 0.5 \cdot M_a)$. Det bakomliggande samplingsavståndet är dubbelt så stort i b) som i a), $\Delta_b = 2 \cdot \Delta_a$. Därför är den maximala spatiella frekvensen hälften så stor i b) som i a), $1/(2\Delta_b) = 0.5 \cdot 1/(2\Delta_a)$. Detta syns ju också tydligt i fouriertransform-bilderna. Antalet sampelpunkter i fourierdomänen (N, M) påverkar alltså inte den maximala spatiella frekvensen utan bara hur tätt fourierdomänen är samplad.



Figur 3.7. Samband mellan samplad kontinuerlig fouriertransform och DFT.

3.5 2D-samband för separabla funktioner

Vi har tidigare påpekat att den tvådimensionella fouriertransformen är separabel (3.3). Om dessutom funktionen som ska transformeras är separabel, dvs kan skrivas på formen $f(x, y) = g(x) \cdot h(y)$ så blir fouriertransformen speciellt enkel. Följande samband gäller:

$$\begin{aligned} F(u, v) &= \mathcal{F}_2[f(x, y)] = \mathcal{F}_x[\mathcal{F}_y[g(x) \cdot h(y)]] = \mathcal{F}_x[g(x) \cdot \mathcal{F}_y[h(y)]] = \\ &= \mathcal{F}_x[g(x)] \cdot \mathcal{F}_y[h(y)] = G(u) \cdot H(v), \end{aligned} \quad (3.14)$$

dvs även den transformerade funktionen är separabel och kan beräknas med två endimensionella transformeringar med efterföljande multiplikation. I Tab. 3.3 visas några matnyttiga funktioner som är separabla.

	Spatialdomän	Fourierdomän
Separabel funktion:	$f(x, y) = g(x) \cdot h(y)$	$F(u, v) = G(u) \cdot H(v)$
Dirac-puls:	$\delta(x, y) = \delta(x) \cdot \delta(y)$	1
Spikmatta:	$\frac{1}{\Delta^2} \text{III}\left(\frac{x}{\Delta}, \frac{y}{\Delta}\right) = \frac{1}{\Delta} \text{III}\left(\frac{x}{\Delta}\right) \cdot \frac{1}{\Delta} \text{III}\left(\frac{y}{\Delta}\right)$	$\text{III}(\Delta u, \Delta v) = \text{III}(\Delta u) \cdot \text{III}(\Delta v)$
Box:	$\Pi(x, y) = \Pi(x) \cdot \Pi(y)$	$\text{sinc}(u) \cdot \text{sinc}(v)$
Böjd pyramid:	$\Lambda(x, y) = \Lambda(x) \cdot \Lambda(y)$	$\text{sinc}^2(u) \cdot \text{sinc}^2(v)$
Gauss:	$e^{-\pi(x^2+y^2)} = e^{-\pi x^2} \cdot e^{-\pi y^2}$	$e^{-\pi(u^2+v^2)} = e^{-\pi u^2} \cdot e^{-\pi v^2}$

Tabell 3.3. Vanliga 2D-funktioner som är separabla och deras fouriertransformer.

3.6 2D-samband för rotationssymmetriska funktioner

Tvådimensionella system uppvisar ofta cirkulär symmetri (= rotationssymmetri, rotationsinvarians). Sålunda är ju optiska system nästan alltid uppbyggda av cirkulära linser. Det är här rimligt att misstänka att systemets punktspridningsfunktion är cirkulärsymmetrisk. Följande teorem sammanfattar vad vi här behöver veta om cirkulärsymmetriska system.

Teorem för fouriertransform av rotationssymmetrisk funktion

Om en tvådimensionell funktion är cirkulärsymmetrisk, dvs

$$f(x, y) = g(r) \quad \text{där} \quad r = \sqrt{x^2 + y^2},$$

så är dess tvådimensionella fouriertransform också cirkulärsymmetrisk, dvs

$$F(u, v) = G(\rho) \quad \text{där} \quad \rho = \sqrt{u^2 + v^2}.$$

I Tab. 3.4 visas några funktioner och deras fouriertransformer. Lägg märke till att Gauss-funktionen är både separabel och rotationssymmetrisk. Funktionen $J_1(\cdot)$ är en så kallad Besselfunktion av första ordningen och $J_1(\pi\rho)/2\rho$ är ganska lik en sinc-funktion.

	Spatialdomän	Fourierdomän
Funktion:	$g(r) = f(x, y)$ $r = \sqrt{x^2 + y^2}$	$G(\rho) = F(u, v)$ $\rho = \sqrt{u^2 + v^2}$
Gauss:	$e^{-\pi r^2}$	$e^{-\pi \rho^2}$
Cylinder:	$\Pi(r)$	$J_1(\pi\rho)/2\rho$
Vulkan:	$1/r$	$1/\rho$

Tabell 3.4. Rotationssymmetriska funktioner och deras fouriertransformer.

3.7 2D faltning och 2D faltningskärnor

Två-dimensionell faltning definieras självfallet som en utvidgning av en-dimensionell faltning,

$$\begin{aligned}
 g(x, y) = (h * f)(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - \alpha, y - \beta) \cdot f(\alpha, \beta) \, d\alpha d\beta \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\alpha, \beta) \cdot f(x - \alpha, y - \beta) \, d\alpha d\beta.
 \end{aligned} \tag{3.15}$$

Den ena av ingångsfunktionerna ska först "vikas" (tyska: falten, engelska: convolve) vilket framgår av det negativa tecknet på integrationsvariablerna (α, β) . I två dimensioner är detta liktydigt med att vi roterar h - (eller f)-funktionen 180°. Därefter låter vi den glida, translateras över f - (eller h)-funktionen. Translationsläget är (x, y) . Se Fig. 3.8. För varje sådant läge beräknas integralen (3.15) och resultatet placeras på koordinat (x, y) i funktionen $g(x, y)$.

Bilder är liksom alla i verkligheten förekommande signaler begränsade. Av Fig. 3.8 inses tydligt att storleken av definitionsområdet på faltningsskärnan h och ingångsbilden f bestämmer definitionsområdet för g . En förutsättning för att Fig. 3.8 skall motsvara verkligheten är att $h(\alpha, \beta)$ är 0 utanför ett område som i Fig. 3.8 antages vara en rektangel

$$h(\alpha, \beta) = 0 \quad \text{för} \quad |\alpha| > \frac{a}{2} \quad \text{eller} \quad |\beta| > \frac{b}{2}.$$

Om nu

$$f(\alpha, \beta) = \text{def} \quad \text{för} \quad 0 < |x| < \frac{A}{2}, \quad 0 < |y| < \frac{B}{2}$$

så resulterar detta uppenbarligen i att $g(x, y)$ inte kan räknas ut för samma bildyta $A \times B$ som definierar $f(\alpha, \beta)$. I ytterkanten av rektangeln $A \times B$ hamnar faltningsskärnan utanför f -funktionen så att

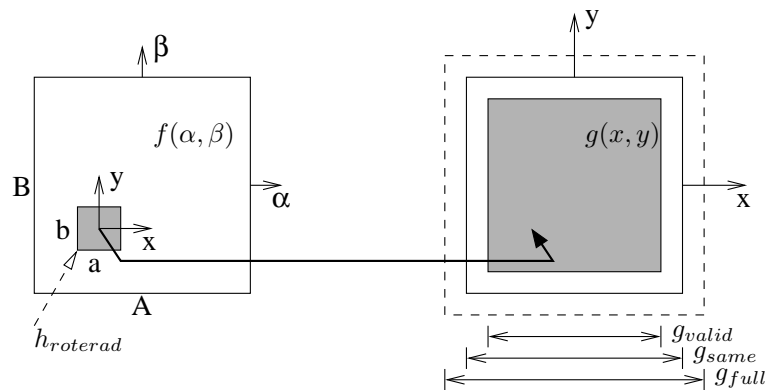
$$g(x, y) = \text{odef} \quad \text{för} \quad |x| > \frac{A-a}{2}, \quad |y| > \frac{B-b}{2}.$$

Om istället $f(\alpha, \beta) = 0$ för $|\alpha| > \frac{A}{2}$ eller $|\beta| > \frac{B}{2}$ så får vi en resultatbild $g(x, y)$ för vilken gäller

$$g(x, y) = 0 \quad \text{för} \quad |x| > \frac{A+a}{2}, \quad |y| > \frac{B+b}{2},$$

dvs bilden g är "större" än ingångsbilden f . I Fig. 3.8 har vi markerat detta fall med g_{full} och det förra fallet med g_{valid} . Ofta vill man ha samma storlek på utgångsbild som på ingångsbild enligt g_{same} . Bilden kan beräknas på samma sätt som för g_{valid} , förutom att de yttre värdena ignoreras. I det fall man inte kan förutsätta att $f(x, y) = 0$ utanför bildytan, men ändå önskar en utgångsbild med samma storlek som ingångsbilden, kan man göra på olika sätt. Sätts värdena i $f(x, y)$ till 0 utanför bildytan så fås kanteffekter i form av t ex en mörk ram runt utgångsbilden. Situationen kan förbättras om värdena i $f(x, y)$ extrapoleras utanför bildytan. En möjlighet är att extrapolera yttersta pixel-lagret, en annan möjlighet är att spegla bilden och lägga till vid ytterkanterna före faltning, dvs

$$f(x, y) = \begin{cases} f(A-x, y), & A/2 < x < A, & -B/2 < y < B/2 \\ f(-A-x, y), & -A < x < -A/2, & -B/2 < y < B/2, \\ f(x, B-y), & -A/2 < x < A/2, & B/2 < y < B, \\ f(x, -B-y), & -A/2 < x < A/2, & -B < y < -B/2, \\ f(A-x, B-y), & A/2 < x < A, & B/2 < y < B \\ f(-A-x, B-y), & -A < x < -A/2, & B/2 < y < B \\ f(A-x, -B-y), & A/2 < x < A, & -B < y < -B/2 \\ f(-A-x, -B-y), & -A < x < -A/2, & -B < y < -B/2 \end{cases}. \quad (3.16)$$



Figur 3.8. Två-dimensionell faltning $g = f * h$ sker med den vikta (ett halvt varv roterade) faltningsskärnan $h_{roterad}$. Beroende på hur man betraktar värdena utanför ingångsbilden får utgångsbildens storlek till g_{valid} , g_{same} eller g_{full} .

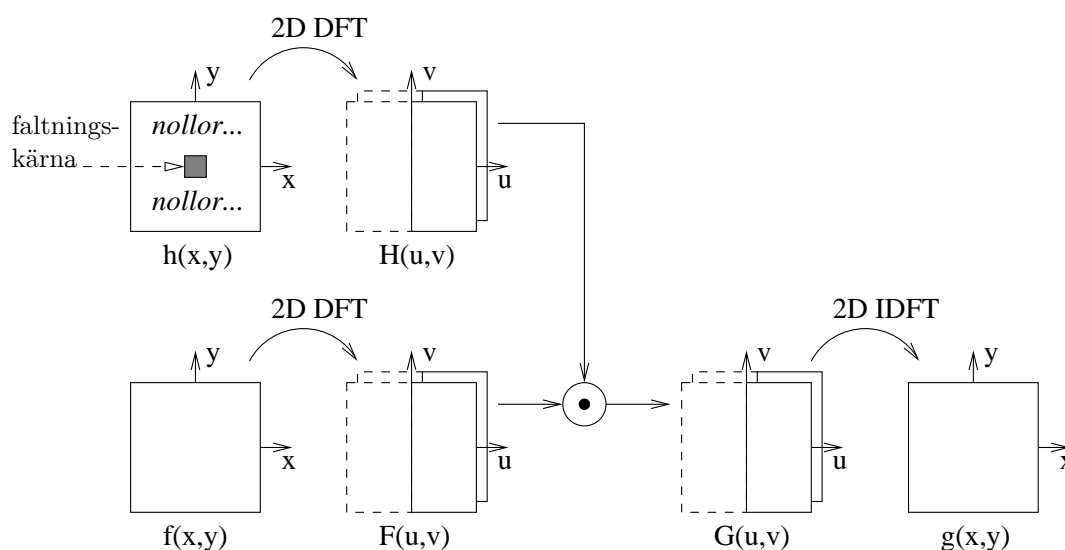
Med diskretiserade signaler, digitala bilder, finns två definitioner på två-dimensionell faltning att välja mellan, dels **linjär diskret faltning**,

$$g(x, y) = (h * f)(x, y) = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} h(x - \alpha, y - \beta) f(\alpha, \beta), \quad (3.17)$$

och dels **cirkulär faltning**,

$$g(x, y) = (h *_N f)(x, y) = \sum_{\alpha=0}^{N-1} \sum_{\beta=0}^{M-1} h(x - \alpha, y - \beta)_N f(\alpha, \beta), \quad (3.18)$$

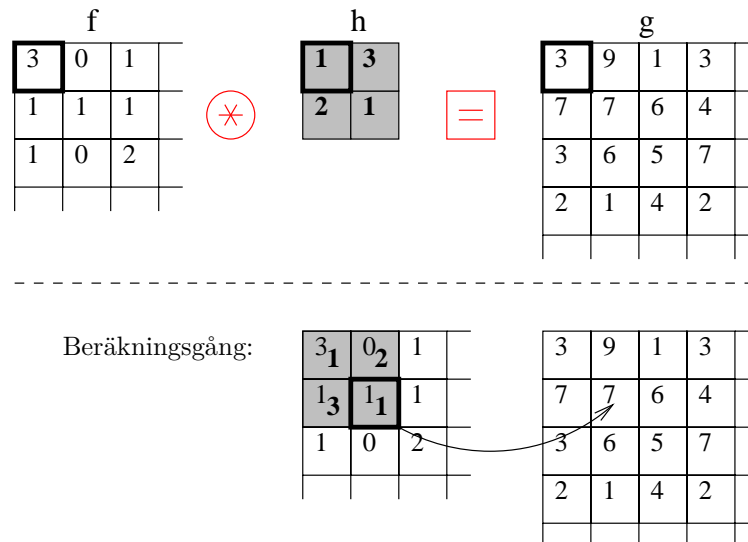
där $(\)_N$ betecknar modulo N operation, dvs $h(\)$ kan uppfattas som periodiskt upprepad eller cirkulär. Notera att vi, trots att det rör sig om diskreta signaler, har valt att kalla variablerna x, y istället för som tidigare n, m . Formeln för linjär diskret faltning är applicerbar på vad vi hittills har sagt i anslutning till Fig. 3.8. Formel för cirkulär diskret faltning gäller även då filtrering utförs genom multiplikation i fourierdomänen och fouriertransformerna utförs med DFT, se Fig. 3.9. Då levereras inte en utgångsbild med förminskat eller förstort definitionsområde, utan utgångsbilden har samma storlek som ingångsbilden. Notera att den lilla faltningskärnan $h(x, y)$ måste paddas med nollor (engelska: zero-padding) för att få en fouriertransform $H(u, v)$ med samma upplösning som $F(u, v)$. Eftersom $h(\)$ (eller $f(\)$) kan uppfattas som periodiskt upprepad eller cirkulär, kommer faltningsresultatet i närheten av bildens kanter påverkas. Utgångsbildens vänsterkant kommer att påverkas av ingångsbildens högerkant. På samma sätt kommer högerkant att påverka vänsterkant och överkant (underkant) att påverka underkant (överkant).



Figur 3.9. Filtrering via multiplikation i DFT-domänen ger cirkulär faltning.

Betrakta linjär diskret faltning i ekvation (3.17). Den ena av ingångsfunktionerna ska först “vikas” (tyska: falten, engelska: convolve) vilket framgår av det negativa tecknet på integrationsvariablerna (α, β) . I två dimensioner är detta liktydigt med att vi roterar h -funktionen 180° . Därefter låter vi den glida, translateras, över f -funktionen. Translationsläget är (x, y) . För varje sådant läge beräknas summan (3.15) och resultatet placeras på koordinat (x, y) i funktionen $g(x, y)$. I Fig. 3.10 visas ett exempel på **2D linjär diskret faltning** enligt ekvation (3.17). Origo har markerats med en tjockare ram.

Faltning sker oftast som i Fig. 3.10 som en operation av en relativt kort sekvens (sekvensen h) på en lång, potentiellt oändligt stor bild. I ett sådant fall kallas funktionen h **faltningskärna**, **faltningsoperator**, **linjär operator**, **filterkärna** eller kort och gott **filter**. Till vänster i Fig. 3.12 och Fig. 3.17 visas hur en faltningskärna kan separeras i ett antal mindre faltningskärnor. Det är lätt att inse att den större faltningskärnan kan erhållas genom att falta ihop de mindre faltningskärnorna. En fördel med att separera en faltningskärna är att beräkningsbördan ofta blir mindre då.

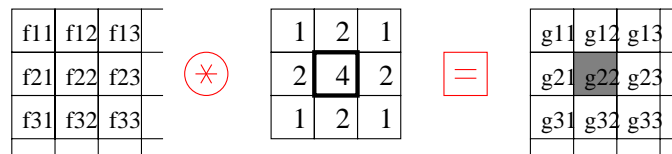


Figur 3.10. Illustration av faltning, $g = h * f$.

I Fig. 3.11 visas en faltningsberäkning mellan en bild f och en faltningskärna. Det gäller för beräkningen av pixeln g_{22} i utgångsbilden g att

$$g_{22} = f_{11} + 2 \cdot f_{12} + f_{13} + \dots + 2 \cdot f_{21} + 4 \cdot f_{22} + 2 \cdot f_{23} + f_{31} + 2 \cdot f_{32} + f_{33}. \quad (3.19)$$

Följaktligen kräver denna faltningskärna 5 multiplikationer och 8 additioner per pixel.

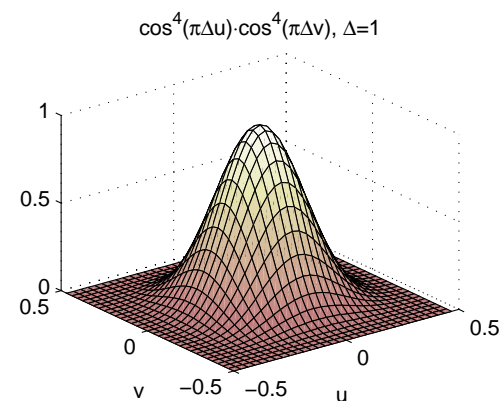
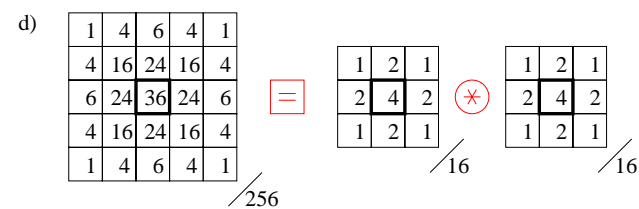
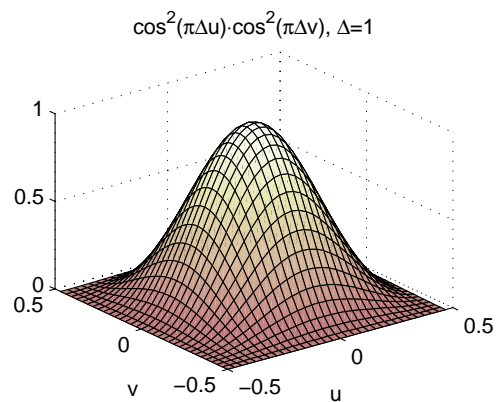
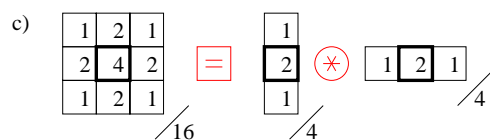
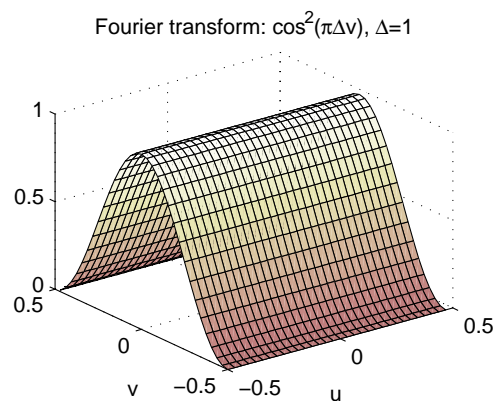
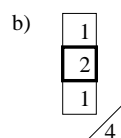
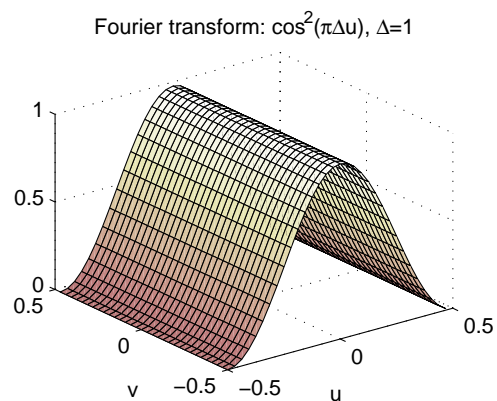
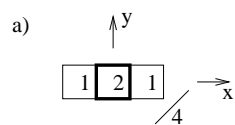


Figur 3.11. Denna faltningskärna kräver 5 multiplikationer och 8 additioner per pixel. Detta inses genom att betrakta ekvation (3.19) där värdet g_{22} beräknas.

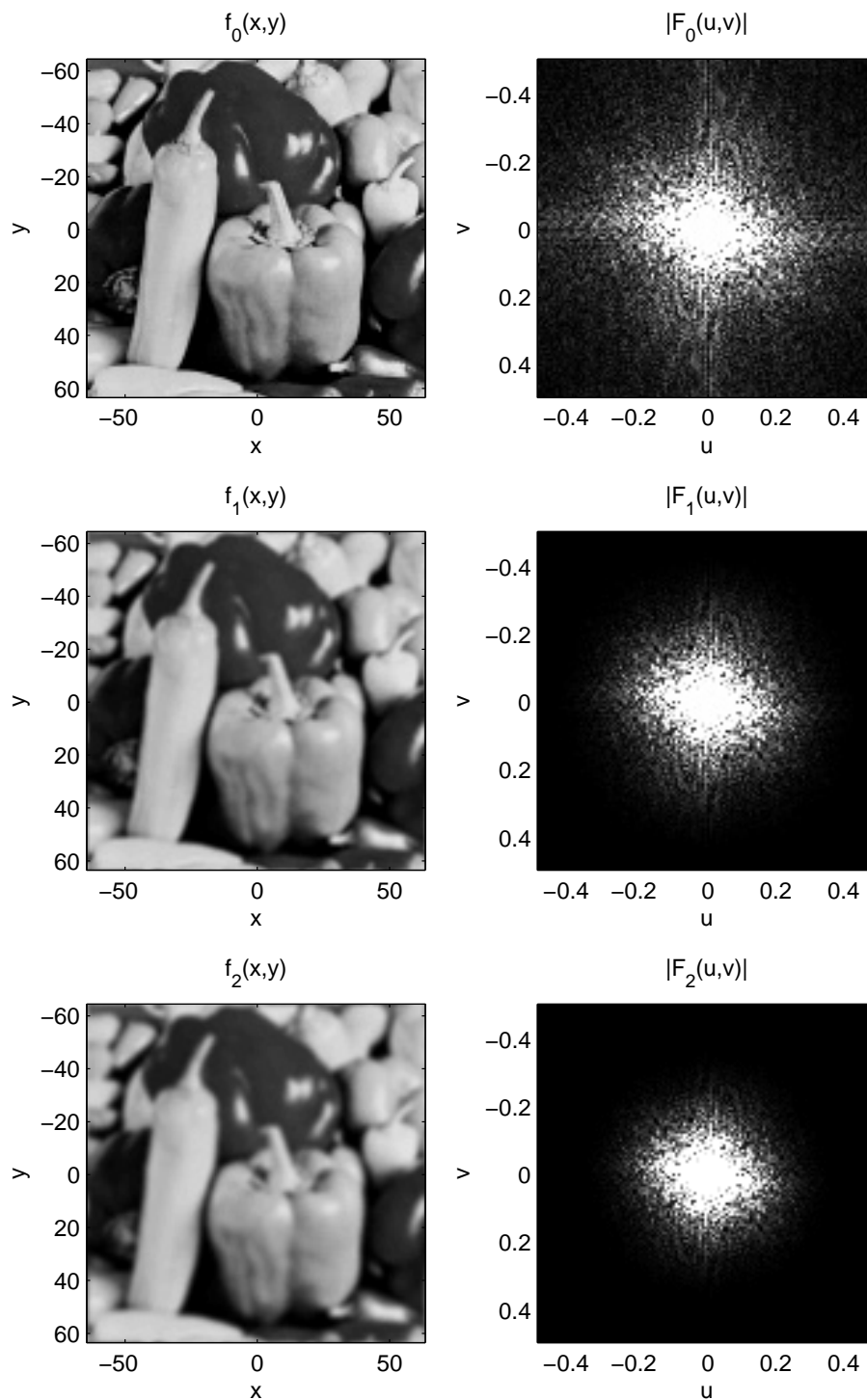
3.8 2D faltningskärnor för lågpasfiltrering

I Fig. 3.12 visas ett antal lågpas-filtrerande faltningskärnor. I Fig. 3.12a visas en faltningskärna som beräknar ett viktat medelvärde under det att den faltas på en bild. Om dess fouriertransform studeras syns det att den är **lågpassfiltrerande i x-led**. Den släpper igenom de låga frekvenser i x-led (små u -värden) och dämpar höga frekvenser i och med att den har värden nära 0 för stora absolutvärden på u . Den faltningskärna som visas i Fig. 3.12b är på samma sätt **lågpassfiltrerande i y-led**.

Kombineras dessa båda faltningskärnor fås den som visas i Fig. 3.12c. Dess fouriertransform är en multiplikation av de båda föregående och den är **lågpassfiltrerande i både x- och y-led**. Den släpper igenom de låga frekvenser (små u - och v -värden) och dämpar höga frekvenser i och med att den har värden nära 0 för stora absolutvärden på u - och v . Fig. 3.12d visas en faltningskärna som kan separeras i två sådana faltningskärnor som visades i Fig. 3.12c. Denna faltningskärna är ännu kraftigare lågpasfiltrerande i både x - och y -led. Upprepad faltning med hjälp av faltningskärnan i Fig. 3.12c ger alltså mer och mer lågpasfiltrering. I Fig. 3.13 visas ett exempel på lågpasfiltrering av en bild med filtret i Fig. 3.12c en och två gånger. Notera att lågpasfiltrets effekt blir en "suddning av" bilden. Notera också att de låga frekvenskomponenterna i fouriertransformen bevaras, medan de höga dämpas.



Figur 3.12. Faltningskärnor för lågpasfiltrering med tillhörande fouriertransform.



Figur 3.13. Lågpassfiltrering. Bilden $f_0(x,y)$ är originalbilden och $|F_0(u,v)|$ är absolutvärdet av dess fouriertransform. Bilden $f_1(x,y)$ är lågpassfiltrerad med filtret i Fig. 3.12c. Bilden $f_2(x,y)$ är lågpassfiltrerad 2 gånger med filtret i Fig. 3.12c, dvs filtret i Fig. 3.12d. (Originalbilden finns på <http://sipi.usc.edu/database/>, University of Southern California.)

3.9 Faltningkärnor för derivering

Analys av lokala variationer i en två-dimensionell signal för med sig behovet av att uppskatta **derivator**, vilket är huvudämnet i detta avsnitt. En faltningkärna som beräknar derivatan kallas en **derivationsoperator**. Vi börjar med att diskutera beräkningen/estimeringen av derivatan i det en-dimensionella fallet. Se Fig. 3.14, som återknyter till Fig. 2.13, grunden för formulering och bevis av samplingsteoremet. Om $f(x)$ är den bakomliggande kontinuerliga signalen och bandbegränsad till $F(u) = 0$ för $|u| > 1/(2\Delta)$ och samplad med $(1/\Delta)\text{III}(x/\Delta)$ fås $s(x)$. Samplingsavståndet är alltså Δ . Då kan vi rekonstruera $f(x)$ enligt

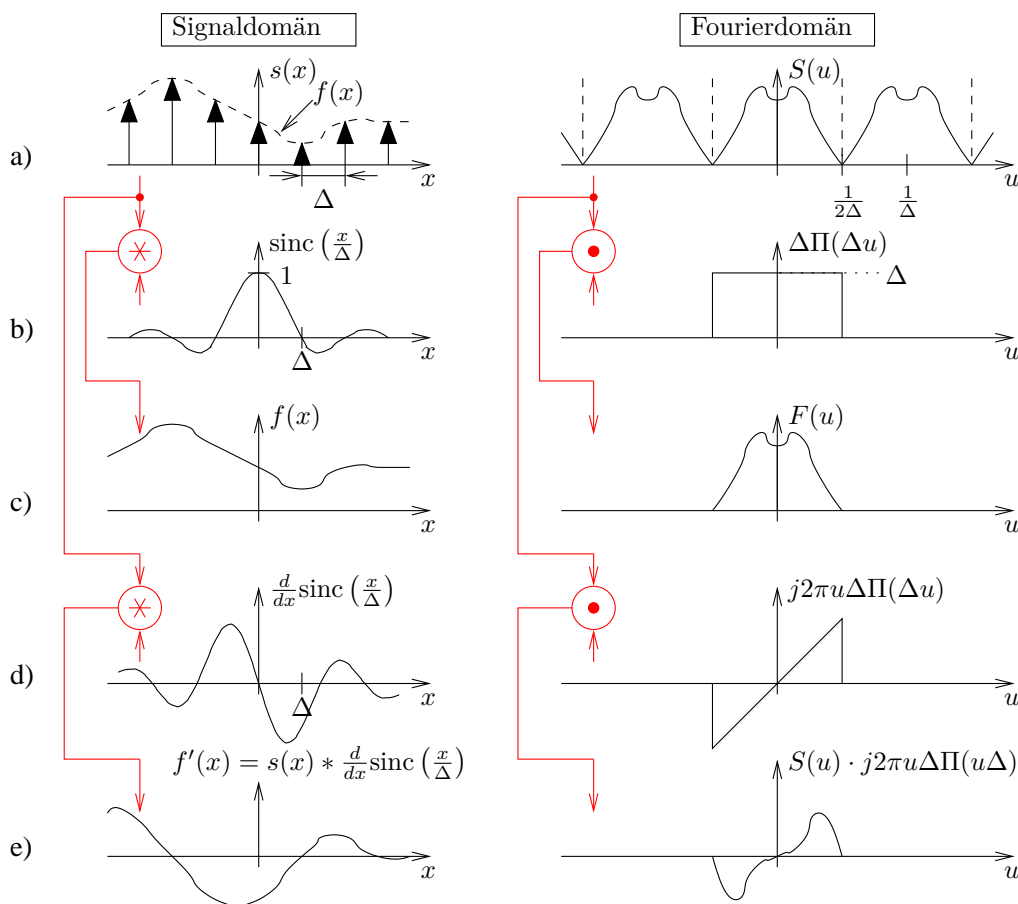
$$f(x) = s(x) * \text{sinc}\left(\frac{x}{\Delta}\right). \tag{3.20}$$

Men om vi kan rekonstruera en deriverbar funktion $f(x)$ ur $s(x)$ kan vi också rekonstruera derivatan $f'(x)$. I fourierdomänen bildas fouriertransformen av derivatan genom multiplikation med $j2\pi u$. Derivatan av rektangelfunktionen $\Delta\Pi(\Delta u)$ blir då $j2\pi u \cdot \Delta\Pi(\Delta u)$ och i signaldomänen blir kombinationen av derivering och rekonstruktion operatoren $g(x)$ enligt (3.20),

$$g(x) = \frac{d}{dx} \text{sinc}\left(\frac{x}{\Delta}\right) \Leftrightarrow j2\pi u \cdot \Delta\Pi(\Delta u). \tag{3.21}$$

Denna funktion visas i Fig. 3.14. Faltad med $s(x)$ resulterar den i derivatan av den bakomliggande kontinuerliga funktionen $f(x)$, dvs

$$f'(x) = \frac{d}{dx} \left[\text{sinc}\left(\frac{x}{\Delta}\right) * s(x) \right] = \frac{d}{dx} \text{sinc}\left(\frac{x}{\Delta}\right) * s(x) = g(x) * s(x). \tag{3.22}$$

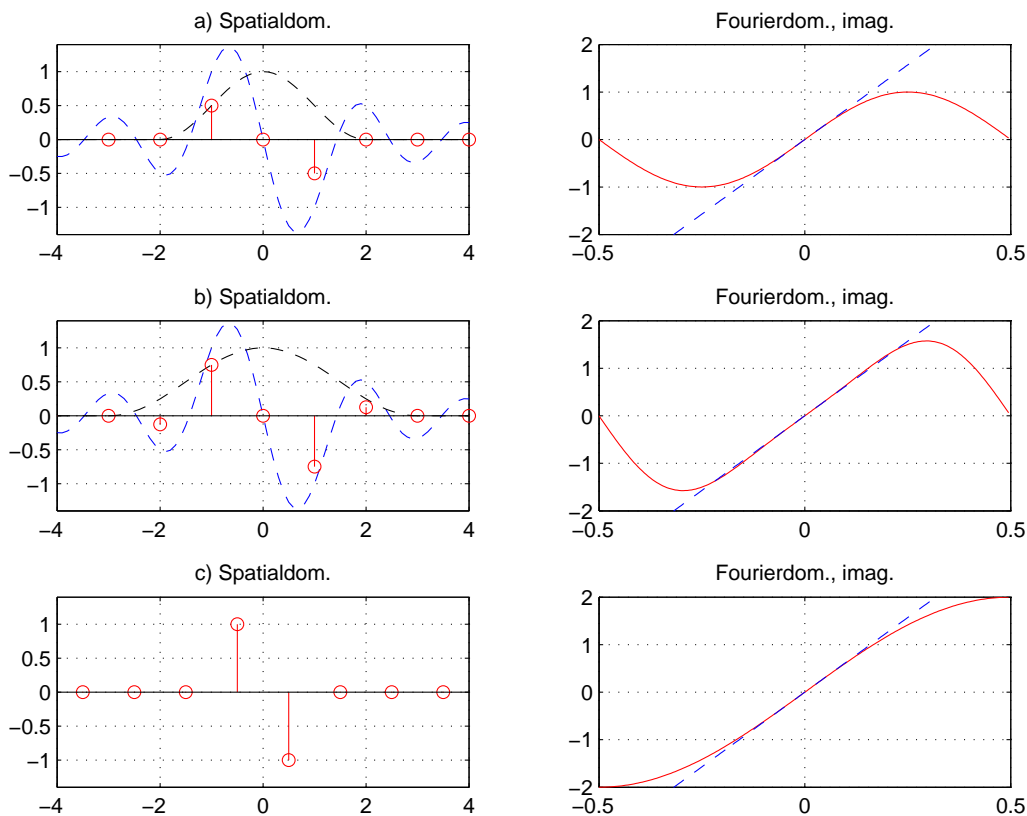


Figur 3.14. På samma sätt som sinc-funktionen rekonstruerar $f(x)$ ur den samplade funktionen $s(x)$, (a,b,c), så rekonstrueras $f'(x)$ ur den samplade funktionen $s(x)$ med hjälp av den ideala derivatorn $g(x) = \frac{d}{dx} \text{sinc}\left(\frac{x}{\Delta}\right)$, (a,d,e).

Den **ideala deriveringsoperatoren**, derivatorn $g(x)$ har oändlig utsträckning. I praktiken måste vi använda en ändlig approximation $\tilde{g}(x)$. Om vi dessutom endast önskar $f'(x)$ i de tidigare använda sampelpunkterna kan vi som $\tilde{g}(x)$ använda en diskret samplad form. Tre sådana approximativa derivatorer (estimatorer, estimerare av 1:a derivatan) visas i Fig. 3.15 där vi har satt $\Delta = 1$. Kärnan a) $\{1\ 0\ -1\}/(2\Delta)$ är enklast och kärnan b) $\{-1\ 6\ 0\ -6\ 1\}/(8\Delta)$ överensstämmer bäst med den ideala funktionen. Dessa har skapats genom att multiplicera $(d/dx)\text{sinc}(x/\Delta)$ med en \cos^2 -funktion som går ner till 0 vid önskad storlek på faltningskärnan. Båda dessa funktioner visas streckade i figurerna. Kärnan c) $\{1\ -1\}/(\Delta)$ är också enkel, men blir utgångsbilden blir förskjuten ett halvt sampelavstånd jämfört med ingångsbilden, vilket är något som programmeraren måste ha kontroll på. Det är inte självklart vilken operator som är bäst. a) $\{1\ 0\ -1\}/(2\Delta)$ kan vara att föredra om man medvetet vill undertrycka inflytande från högre frekvenser inom bandbreddsområdet. Det senare är ett inte ovanligt önskemål eftersom signal/brusförhållandet ofta är dåligt i de högre frekvensområdena. Man brukar uttrycka detta så att ideal derivering inte alltid är optimal därför att den förstärker bruset i bilden. Som framgår av Fig. 3.15 har alla faltningskärnorna det gemensamt att de för låga frekvenser u överensstämmer väl med den ideala $G(u) = j2\pi u \cdot \Pi(u\Delta)$ som visas streckad i figuren. Det bör också observeras att det samplade resultatet av $f'(x)$, dvs

$$\left(\frac{1}{\Delta}\right) \text{III}\left(\frac{x}{\Delta}\right) \cdot f'(x) \approx \left(\frac{1}{\Delta}\right) \text{III}\left(\frac{x}{\Delta}\right) \cdot \tilde{g}(x) * s(x), \quad x = \dots -2, -1, 0, 1, 2 \dots$$

i fourierdomänen har ett repetitivt utseende med perioden $1/\Delta$.



Figur 3.15. Tre approximationer av den ideala deriveringsoperatoren. Streckad linje $=j2\pi u$.

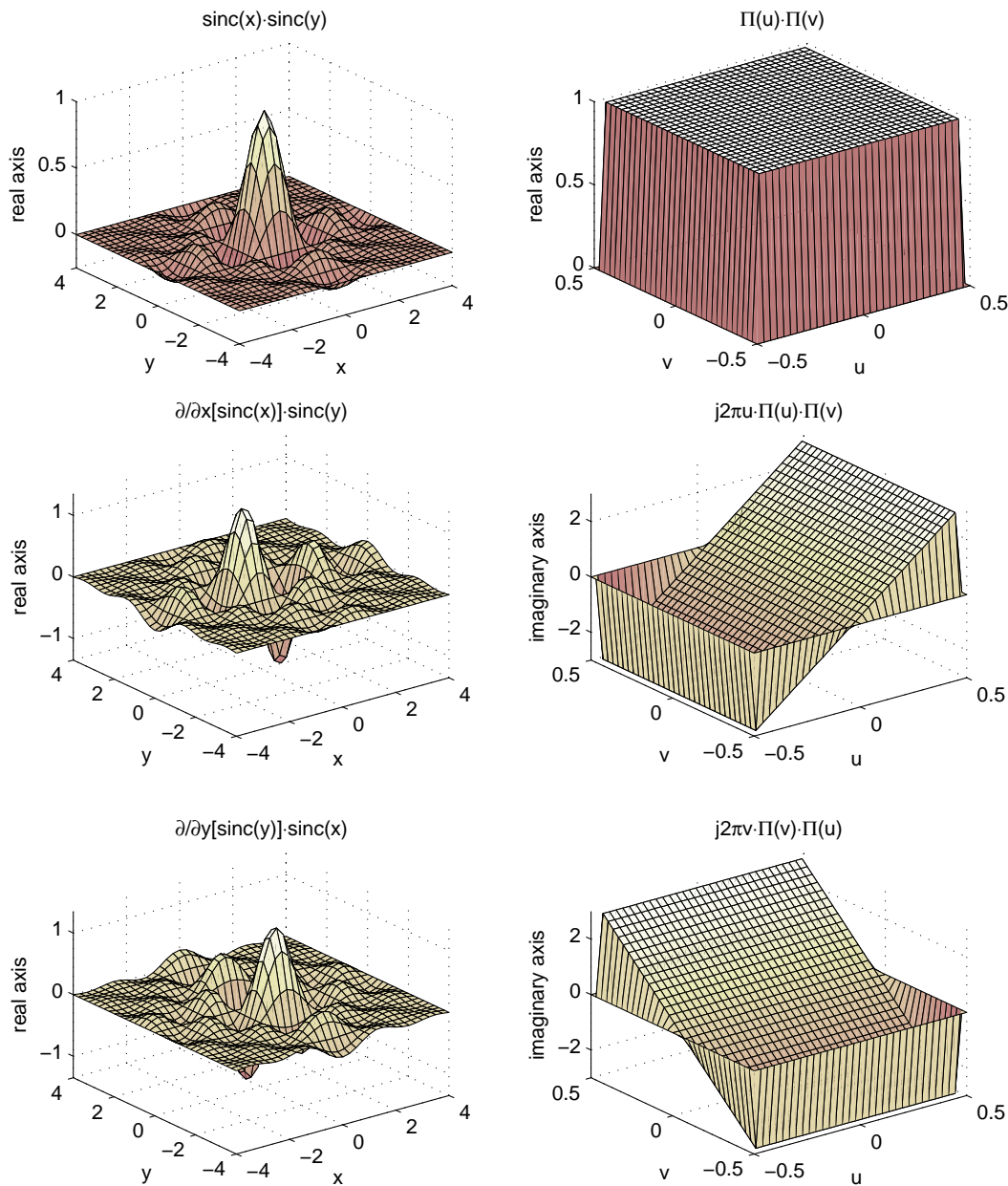
För att kunna diskutera **två-dimensionell derivering** bör vi först klargöra hur vi rekonstruerar en kontinuerlig bildsignal ur en samplad version. Perfekt rekonstruktion av en bandbegränsad signal kan ske i fourierdomänen med den två-dimensionella rektangelfunktionen som visas i i Fig. 3.16 som vi skriver som

$$G_0(u, v) = \Delta\Pi(\Delta u) \cdot \Delta\Pi(\Delta v) = \Delta^2\Pi(\Delta u, \Delta v), \tag{3.23}$$

där Δ är samplingsavståndet av den bakomliggande två-dimensionella kontinuerliga funktionen. Liksom i det en-dimensionella fallet kan vi, åtminstone teoretiskt, utföra rekonstruktionen i signaldomänen, bildplanet, med

$$g_0(x, y) = \text{sinc}\left(\frac{x}{\Delta}\right) \cdot \text{sinc}\left(\frac{y}{\Delta}\right). \tag{3.24}$$

Funktionerna $g_0(x, y) = \text{sinc}(x) \cdot \text{sinc}(y)$ och $G_0(u, v) = \Pi(u, v)$ visas i Fig. 3.16.



Figur 3.16. Ideala 2D-operatorer för rekonstruktion och derivering i x- och y-led.

Partiell derivering i x -led av en funktion $f(x, y)$ är liktydigt med multiplikation i fourierdomänen med $j2\pi u$. På samma sätt som för det en-dimensionella fallet så multiplicerar vi därför vår rekonstruktionsfunktion $G_0(x, y)$ med denna deriveringsfunktion för att nå den kombinerade effekten av rekonstruktion och derivering av den bakomliggande kontinuerliga funktionen. Resultatet blir i fourierdomänen två stycken utskurna lutande plan, som beskrivs med uttrycken

$$G_x(u, v) = j2\pi u \cdot \Delta^2 \Pi(\Delta u, \Delta v), \quad (3.25)$$

$$G_y(u, v) = j2\pi v \cdot \Delta^2 \Pi(\Delta u, \Delta v). \quad (3.26)$$

Den inversa fouriertransformen av dessa ekvationer, (3.25) och (3.26) blir

$$g_x(x, y) = \frac{\partial}{\partial x} \operatorname{sinc}\left(\frac{x}{\Delta}\right) \cdot \operatorname{sinc}\left(\frac{y}{\Delta}\right), \quad (3.27)$$

$$g_y(x, y) = \frac{\partial}{\partial y} \operatorname{sinc}\left(\frac{y}{\Delta}\right) \cdot \operatorname{sinc}\left(\frac{x}{\Delta}\right). \quad (3.28)$$

De två funktionerna g_x och g_y är de **ideala estimatorerna för 1:a derivatorna** av en bandgränsad två-dimensionell signal. De har ett liknande utseende som den ideala endimensionella derivatorn $g(x)$ i Fig. 3.14. Sammansatta till vektorn (g_x, g_y) utgör de den ideala estimatorn för gradienten. Funktionerna $g_x(x, y)$, $G_x(u, v)$, $g_y(x, y)$ och $G_y(u, v)$ visas i Fig. 3.16 för $\Delta = 1$. Funktionerna g_y och G_y är som synes identiska med g_x och G_x roterade 90° .

Ett antal **approximativa derivatorer** visas av Fig. 3.17 tillsammans med respektive fouriertransformer. I Fig. 3.17a visas en differentierande faltningkärna i x -led. Under det att den faltas på en bild beräknar den differensen mellan framförvarande och bakomliggande pixel. Dess fouriertransform är $j \sin(2\pi \Delta u) / \Delta$. För små värden på u är denna fouriertransform ungefär lika med $j2\pi u$ som enligt Tab. 3.1 motsvarar derivering i x -led. Eftersom fouriertransformen har värden nära 0 för stora absolutvärden på u är denna faltningkärna också i viss mån lågpasfilterande i x -led. Denna faltningkärna är alltså både deriverande och lågpasfilterande i x -led. I Fig. 3.17b visas en differentierande faltningkärna i y -led. Ett liknande resonemang som förut ger att denna kärna är både deriverande och lågpasfilterande i y -led.

Genom att falta en i y -led lågpasfilterande faltningkärna med en i x -led differentierande fås en faltningkärna som brukar kallas **sobel-x**. Den är deriverande i x -led och lågpasfilterande i både x - och y -led. På liknande sätt gäller att faltningkärnan **sobel-y** är deriverande i y -led och lågpasfilterande i både x - och y -led. Sobel-x och Sobel-y kallas med ett gemensamt namn **sobel-operatorer** och de visas i Fig. 3.17 c,d.

Betrakta operatorparet $[D_x(u, v), D_y(u, v)]$ nedan. Det utför en lågpasfiltrering med $H_x(u, v)$ och $H_y(u, v)$ samt derivering i x - och y -led med $j2\pi u$ och $j2\pi v$,

$$\begin{cases} D_x(u, v) &= H_x(u, v) \cdot j2\pi u, \\ D_y(u, v) &= H_y(u, v) \cdot j2\pi v. \end{cases} \quad (3.29)$$

Låt nu $d_x(x, y)$ och $d_y(x, y)$ vara inversfouriertransformerna av $D_x(u, v)$ och $D_y(u, v)$. Kalla faltningssvaren av detta operatorpar på bilden $f(x, y)$ för $f_x(x, y)$ och $f_y(x, y)$, dvs

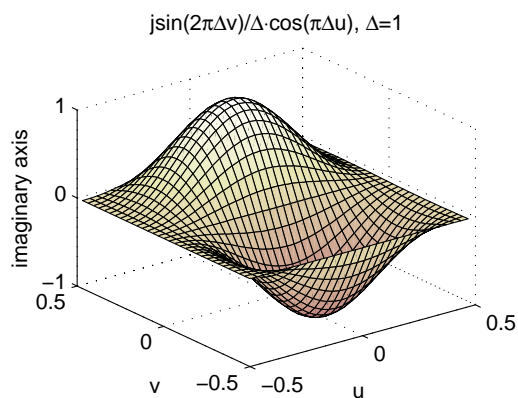
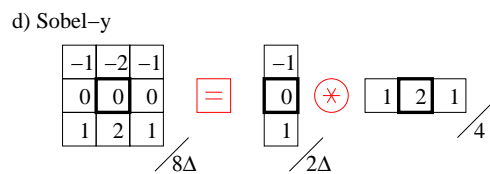
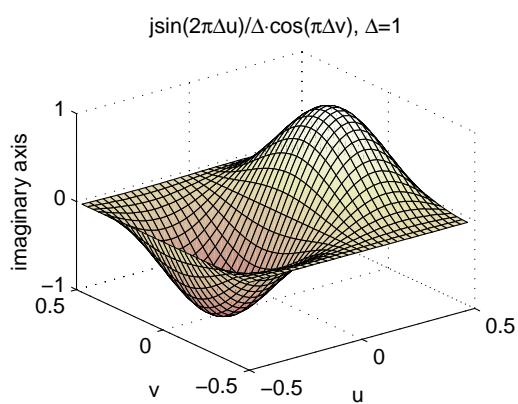
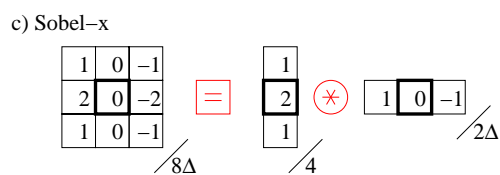
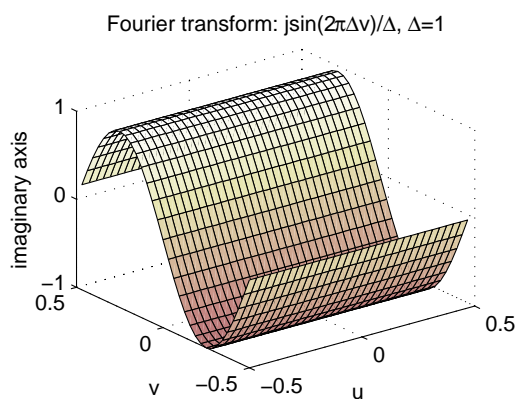
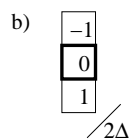
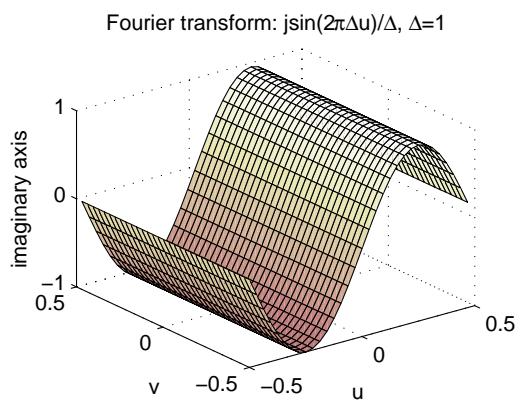
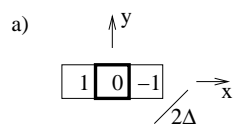
$$\begin{cases} f_x(x, y) &= d_x(x, y) * f(x, y), \\ f_y(x, y) &= d_y(x, y) * f(x, y). \end{cases} \quad (3.30)$$

Sammansatta till en vektor utgör de en estimator för **gradienten** i bilden,

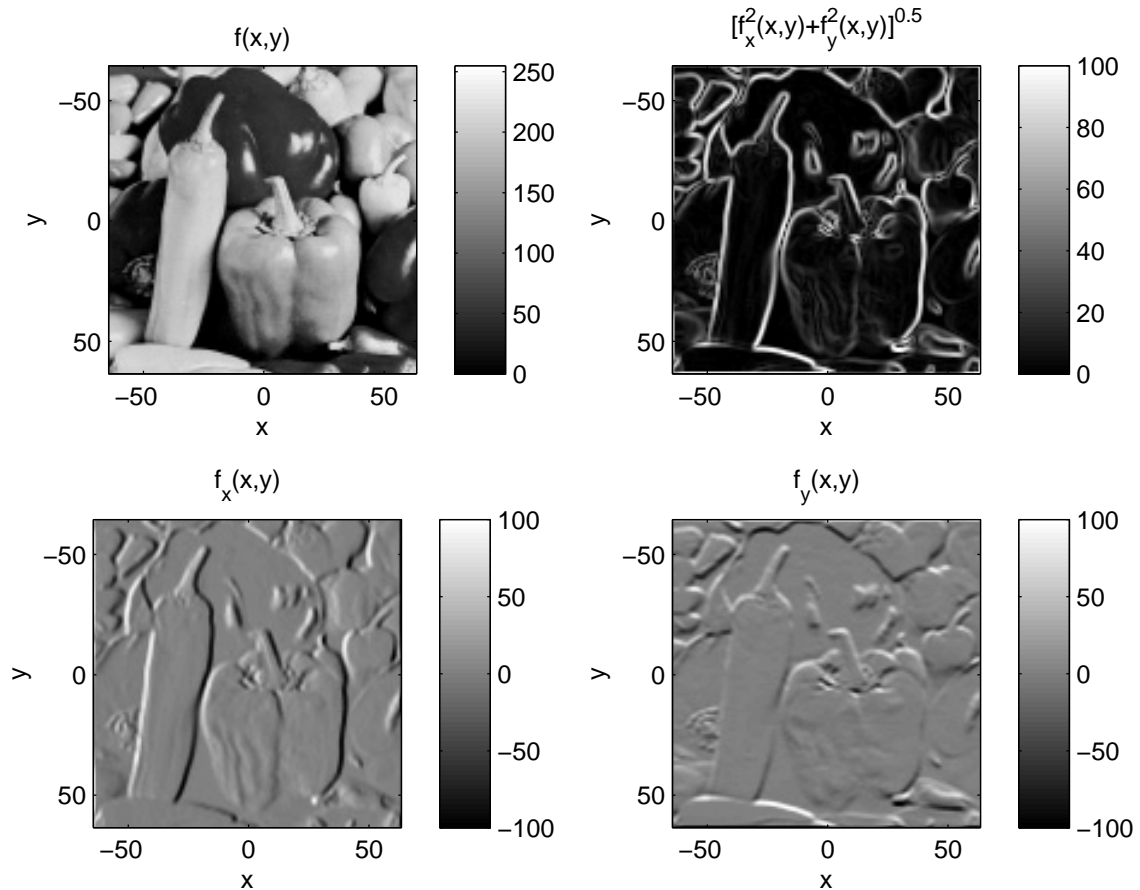
$$\text{gradienten} = \nabla = [f_x(x, y), f_y(x, y)]. \quad (3.31)$$

I Fig. 3.18 visas ett exempel på derivatafiltrering av en bild. Bilden $f(x, y)$ är originalbilden. Derivatabiliden i x -led, $f_x(x, y)$, är faltad med Sobel-x filtret i Fig 3.17c och derivatabilden i y -led, $f_y(x, y)$, är faltad med Sobel-y filtret i 3.17d. Kantdetektering erhålls genom att beräkna magnituden på gradienten,

$$|\nabla| = \sqrt{f_x(x, y)^2 + f_y(x, y)^2}. \quad (3.32)$$



Figur 3.17. Faltningkärnor för derivering med tillhörande fouriertransform. Dessa faltningkärnor ger också en viss lågpas-effekt.

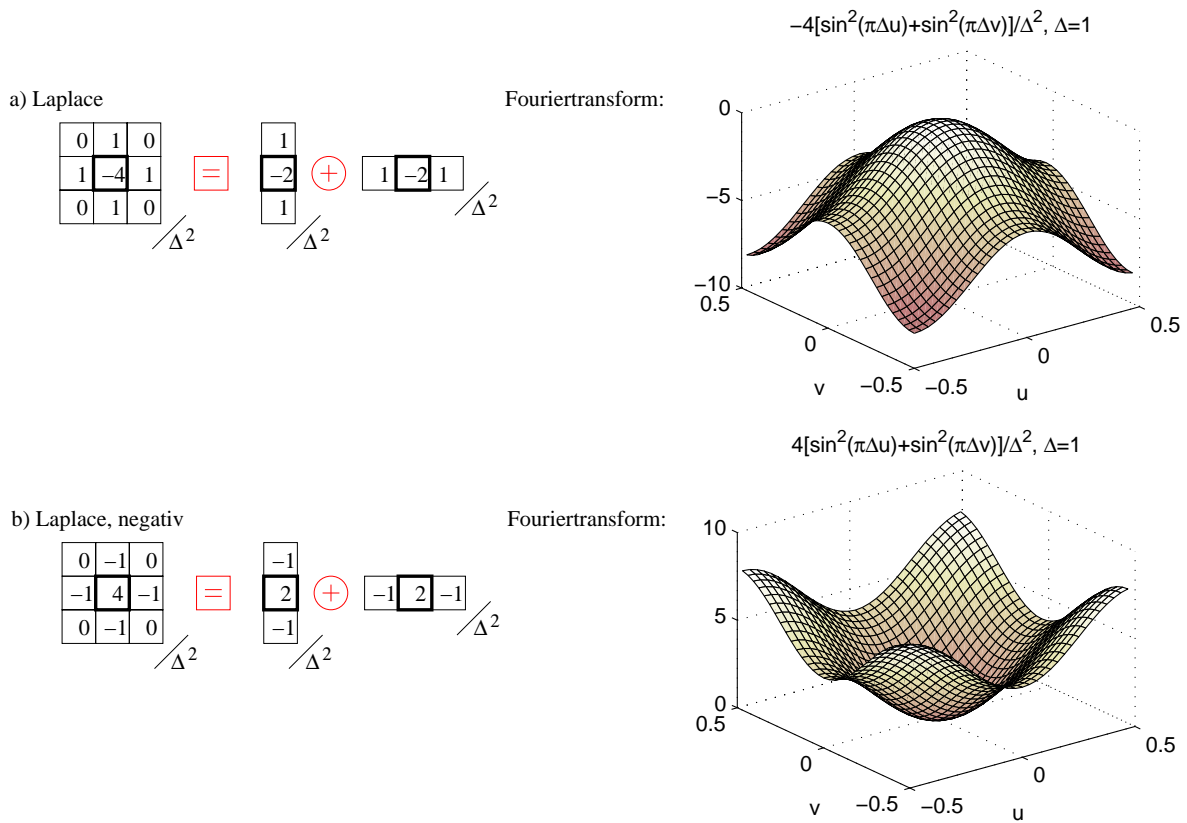


Figur 3.18. Derivering och kantdetektering. Bilden $f(x,y)$ är originalbilden. Derivatabelden i x -led, $f_x(x,y)$, är faltad med Sobel- x filtret. Derivatabelden i y -led, $f_y(x,y)$, är faltad med Sobel- y filtret. Kantdetektering erhålls med hjälp av magnituden på gradienten, $\sqrt{f_x(x,y)^2 + f_y(x,y)^2}$. (Originalbilden finns på <http://sipi.usc.edu/database/>, University of Southern California.)

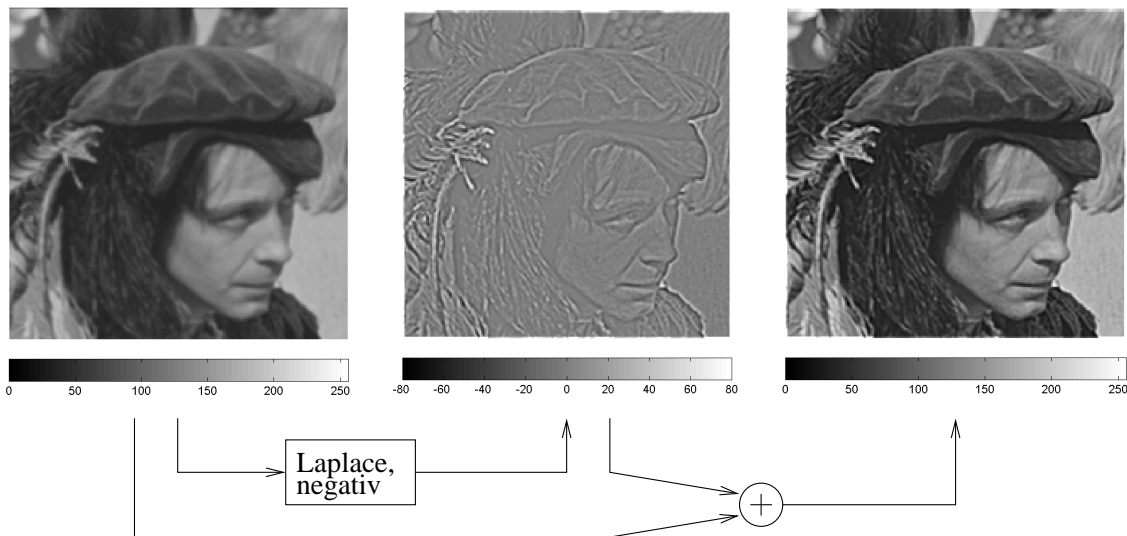
Fig. 3.19a visar en kärna med fouriertransformen $-4(\sin^2(\pi\Delta u) + \sin^2(\pi\Delta v))/\Delta^2$. För små värden på u är denna fouriertransform ungefär lika med $-4\pi(u^2 + v^2)$. Enligt Tab. 3.1 motsvarar detta applikation av laplace-operatorn $\partial^2/\partial x^2 + \partial^2/\partial y^2$. Eftersom fouriertransformen inte följer denna kurva för stora absolutvärden på u och v utan böjer av, är denna faltningskärna också i viss mån lågpas-filtrerande i båda lederna.

Faltningskärnan i Fig. 3.19a är alltså både en **laplace-operator** samt lågpasfiltrerande i x - och y -led. Sammantaget blir detta faktiskt ett högpasfilter i x - och y -led. Det kan vara förvillande att fouriertransformen har negativa värden. Kom då ihåg att om man ska uttala sig om ett filters lågpas-, högpas- eller bandpass-karaktär så bör man betrakta absolutvärdet av filtrets fouriertransform. Man kan dock säga att högpasfiltret har negativt tecken eller att högpasfiltret fasvrider 180 grader. Om man multiplicerar laplace-operatorn med -1 erhålles faltningskärnan i Fig. 3.19b. Denna är som synes ett helt normalt högpas-filter i fourierdomänen.

Slutligen visas i Fig. 3.20 ett exempel på **bildförbättring** genom användning av laplace-filtret. Laplace-bilden erhålls genom att falta originalbilden med $\text{laplace}_{negativ}$ -operatorn i Fig. 3.19. Genom att addera originalbilden och laplacebilden erhålls en bild med tydligare detaljer. Detta beror på att detaljerna innehåller höga frekvenser som förstärks av laplacefiltret.



Figur 3.19. Faltningskärnor med tillhörande fouriertransform. a) Faltningskärna som fungerar som en approximativ laplaceoperator $\partial^2/\partial x^2 + \partial^2/\partial y^2$. b) Med negativt tecken erhålles ett högpasfilter.



Figur 3.20. Användning av laplace-filtrer. Bilden till vänster faltas med $laplace_{negativ}$ -operatorn i Fig. 3.19b. Ett enkelt exempel på bildförbättring erhålles genom att addera bilden till vänster och den $laplace_{negativ}$ -filtrerade bilden. Då erhålles en bild med tydligare detaljer (detaljer = högpassinformation). (Originalbilden finns på <http://sipi.usc.edu/database/>, University of Southern California.)

3.10 Normalisering av faltningskärnor

I Fig. 3.12 har varje lågpas-filtrerande faltningskärna försetts med en normaliseringskonstant så att det lokala medelvärdet bevaras för varje omgivning i bilden, där omgivningens storlek är samma som faltningskärnans. Låt faltningskärnans storlek vara N pixlar och dess koefficienter vara f_1, f_2, \dots, f_N . Då blir normaliseringskonstanten

$$K = \sum_{i=0}^N f_i. \quad (3.33)$$

Betrakta t ex faltningskärnan i Fig. 3.12c. För denna gäller att

$$K = 1 + 2 + 1 + 2 + 4 + 2 + 1 + 2 + 1 = 16. \quad (3.34)$$

Om normaliseringskonstanten sätts till ett för lågt värde kommer bildens medelvärde att öka och bilden blir för ljus. På motsvarande sätt blir bilden för mörk om normaliseringskonstanten sätts till ett för högt värde.

I Fig. 3.17 och Fig. 3.19 har varje faltningskärna försetts med normaliseringskonstanter så att de deriverande kärnorna beräknar korrekt skalade derivator för långsamt varierande gråskalevariationer. Det bakomliggande sampelavståndet är Δ . Om man inte vet sampelavståndet kan man t ex sätta $\Delta = 1$. Normalisering med hänsyn taget till absolutbeloppet (L_1 -norm),

$$K = \sum_{i=0}^N |f_i|.$$

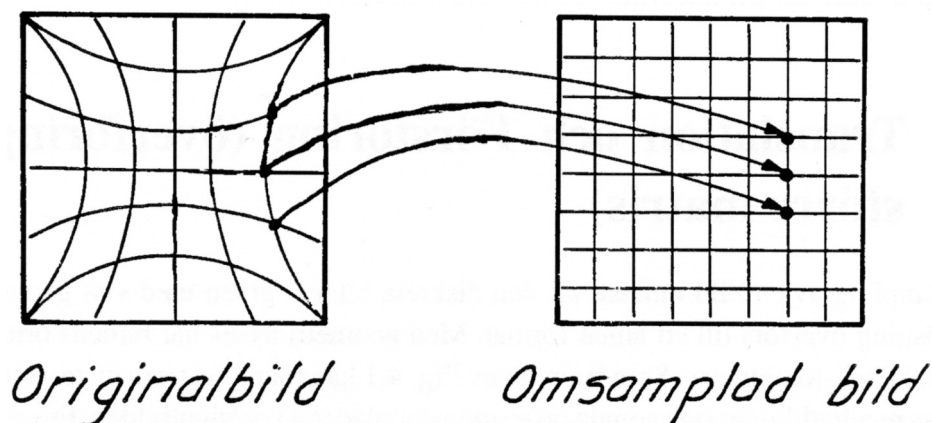
brukar ge ett rimligt värde, dvs omfånget på pixelvärdena i utbilden brukar bli i samma storleksordning som för inbilden. Om man enbart ska använda den den faltade bilden för att titta på kan en annan möjlighet vara att justera normaliseringsfaktorn så att den faltade bildens kontrast blir subjektivt bra.

Kapitel 4

Omsampling av bilder

4.1 Translation och förstoring (uppsampling)

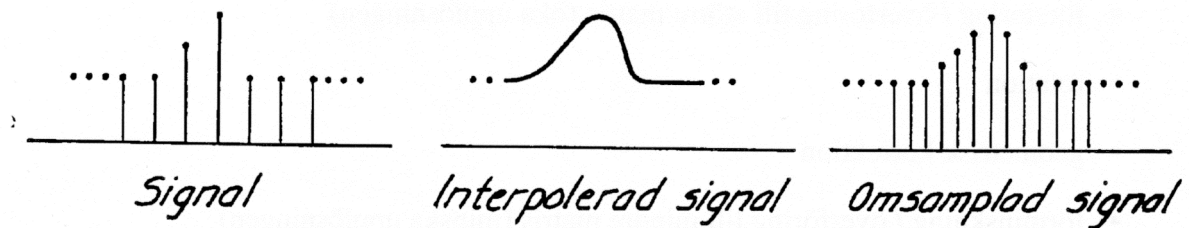
Omsampling av en bild innebär att den diskreta bilden, given med viss geometri och upplösning överförs till ett annat format. Med geometri avses här bildens orientering, eventuella distorsion osv. Som framgår av Fig. 4.1 kan man illustrera distorsion genom att visa resultatbildens ortogonala rastermönster placerat i originalbilden. En satellitbild är sålunda ofta grovt distorderad och behöver korrigeras för detta för att få rätt orientering och samma upplösning som en viss kartbild.



Figur 4.1. Det principiella förfaringssättet vid omsampling: Den nya bildens sampel skall räknas fram ur originalet i ett antal mer eller mindre ordnade nätpunkter som inte sammanfaller med givna originalsampel.

Samtliga fall uppräknade nedan är specialfall av det generella omsamlingsproblemet.

- translation
- förstoring, uppsampling, överföring till större matris, öka upplösningen
- rotation
- geometrisk korrektion
- förminskning, nedsampling, överföring till mindre matris, minska upplösningen



Figur 4.2. Omsampling kan delas upp i två processer: Interpolation av den diskreta signalen så att en kontinuerlig signal erhålles, därefter sampling av denna signal.

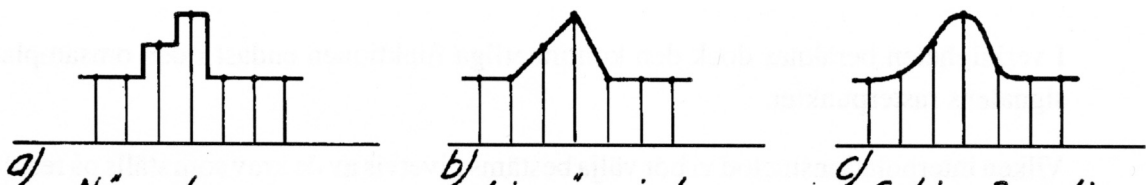
Omsampling kan anses bestå av två processer (se Fig. 4.2).

- **Interpolation** (faltning) av den diskreta signalen med hjälp av någon interpolationsfunktion, vilket resulterar i en kontinuerlig signal.
- **Sampling** av den kontinuerliga signalen.

I verkligheten beräknas dock den kontinuerliga funktionen endast i den omsamplade signalens rasterpunkter.

Vilken interpolationsmetod vi bör välja bestäms givetvis av de krav som ställs på resultat och beräkningstid. Ofta skall en omsamplad bild bearbetas vidare i dator. Det är då inte nödvändigtvis ögats uppfattning av likhet som bör eftersträvas utan kanske hellre en matematisk trohet mot originalet. Vanliga metoder vid omsampling är **närmsta granne** metoden, **linjär interpolation** och interpolation med en funktion av typ **cubic spline**.

Vid den enklaste metoden, närmsta granne, tilldelas en ny bildpunkt värdet av den av originalpunkterna som ligger närmast. Den interpolerade signalen får då ett mycket kantigt utseende (Fig. 4.3 a). Vid linjär interpolation interpoleras linjärt mellan två närliggande punkter (Fig. 4.3 b). Med en viss typ av cubic spline, som använder sig av fyra punkter vid interpolationen, får den kontinuerliga signalen en mjukare form (Fig. 4.3 c). I två dimensioner blir givetvis fler punkter inblandade. Vi betraktar dock tills vidare omsamplingen i endast en dimension.

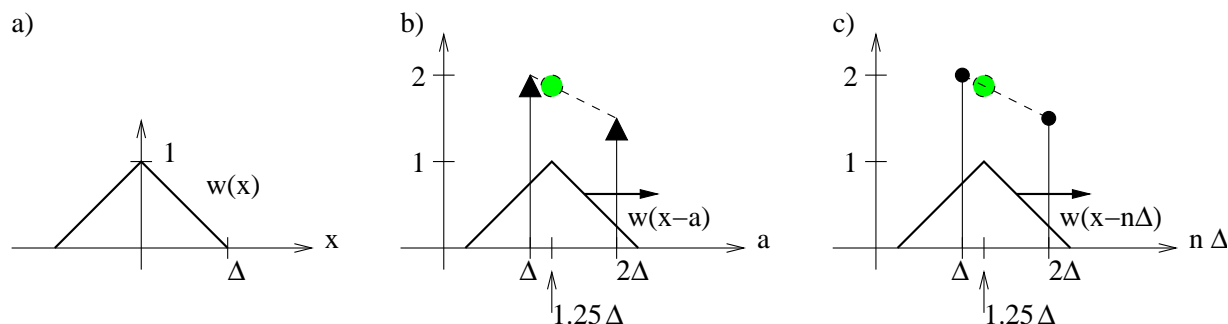


Figur 4.3. Interpolerad signal för olika metoder. a) Närmsta granne. b) Linjär interpolation. c) Cubic spline interpolation.

Vi ska nu komma fram till en metod för interpolation. Låt oss först betrakta faltningen med en interpolationsfunktion i detalj. Låt $s(x) = g(x) \cdot \sum_{n=-\infty}^{\infty} \delta(x - n\Delta)$ vara den samplade insignalen och låt $w(x)$ vara interpolationsfunktionen. Då gäller för faltningen

$$\begin{aligned}
 (w * s)(x) &= \int_{-\infty}^{\infty} \overbrace{w(x-a) \cdot s(x) da}^{\text{faltning med kontinuerligt samplad funktion}} = \int_{-\infty}^{\infty} w(x-a) \cdot \sum_{n=-\infty}^{\infty} g(a) \delta(x-n\Delta) da \\
 &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} w(x-a) \cdot g(a) \delta(x-n\Delta) da = \underbrace{\sum_{n=-\infty}^{\infty} w(x-n\Delta) \cdot g(n\Delta)}_{\text{faltning med diskreta sampelvärden}} \quad (4.1)
 \end{aligned}$$

Faltningen kan alltså beskrivas antingen med integral eller summa. Exempel på framräkning av ett nytt sampelvärde med interpolationsfunktionen $w(x) = \Lambda(x/\Delta)$ (linjär interpolation) visas för båda fallen i



Figur 4.4. Interpolation med $w(x) = \Lambda(x/\Delta)$ (linjär interpolation) för framräkning av ett nytt sampelvärde. a) Interpolationsfunktionen. b) Faltning med kontinuerligt samplad funktion. c) Faltning med diskreta sampelvärden.

Fig. 4.4. I det första fallet gäller

$$\begin{aligned}
 (w * s)(x) &= \int_{-\infty}^{\infty} w(x-a) \cdot s(a) da \\
 &= \int_{-\infty}^{\infty} w(1.25\Delta - a) \cdot 2\delta(a - \Delta) + w(1.25\Delta - a) \cdot 1.5\delta(a - 2\Delta) da \\
 &= \int_{-\infty}^{\infty} w(1.25\Delta - \Delta) \cdot 2\delta(a - \Delta) + w(1.25\Delta - 2\Delta) \cdot 1.5\delta(a - 2\Delta) da \\
 &= w(0.25\Delta) \cdot 2 + w(-0.75\Delta) \cdot 1.5 = \Lambda(0.25) \cdot 2 + \Lambda(-0.75) \cdot 1.5 \\
 &= 0.75 \cdot 2 + 0.25 \cdot 1.5 = 1.875
 \end{aligned} \tag{4.2}$$

och i det senare fallet gäller

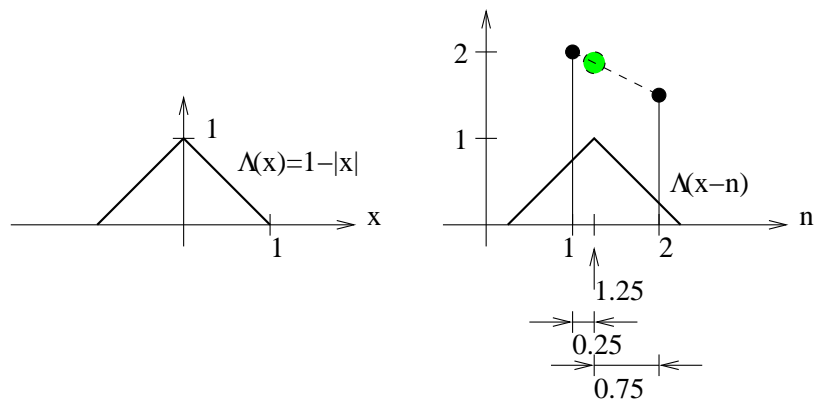
$$\begin{aligned}
 (w * s)(x) &= \sum_{n=-\infty}^{\infty} w(x - n\Delta) \cdot g(n\Delta) \\
 &= w(1.25\Delta - \Delta) \cdot 2 + w(1.25\Delta - 2\Delta) \cdot 1.5 + \\
 &= w(0.25\Delta) \cdot 2 + w(-0.75\Delta) \cdot 1.5 = \Lambda(0.25) \cdot 2 + \Lambda(-0.75) \cdot 1.5 \\
 &= 0.75 \cdot 2 + 0.25 \cdot 1.5 = 1.875.
 \end{aligned} \tag{4.3}$$

Vi är nu redo för en förenklad beskrivning av förfarandet vid interpolation. Som ovan nämndes beräknas inte hela den kontinuerliga signalen. Istället faltas originaldata med interpolationsfunktionen endast i de punkter som behöver beräknas.

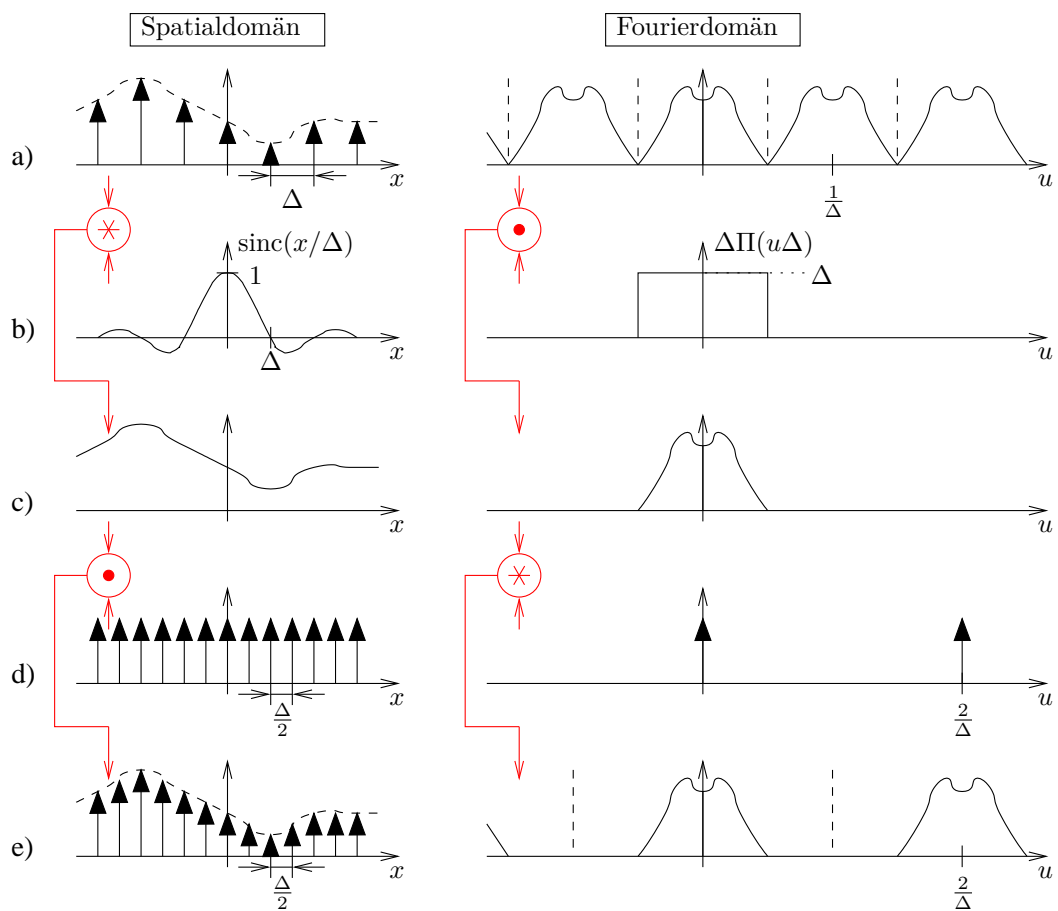
Interpolation, förenklad beskrivning

- Se Fig. 4.5 där vi har satt $\Delta = 1$.
- Flytta interpolationsfunktionen till den önskade positionen.
- Interpolationsfunktioner är jämna \Rightarrow betrakta bara avstånd.
- Vikta sampelvärdena med motvarande höjd på interpolationsfunktionen och summera.
- I det här fallet, med linjär interpolation blir:
Interpolationsvärdet = $\Lambda(0.25) \cdot 2 + \Lambda(0.75) \cdot 1.5 = 0.75 \cdot 2 + 0.25 \cdot 1.5 = 1.875$

Förutsatt att en signal är bandbegränsad och samplad tillräckligt tätt (enligt samplingsteoremet) kan den rekonstrueras i varje punkt genom att i frekvensplanet multiplicera med ett idealt lågpasfilter, en **rektangelfunktion** som sträcker sig fram till bandgränsen. I bildplanet motsvarar detta faltning med en **sinc-funktion**. Fig. 4.6 illustrerar **ideal uppsampling**. Efter uppsamlingen är Fouriertransformen bevarad helt intakt. Den enda skillnaden jämfört med före omsampling är att upprepningarna kommer glesare. De nya högsta frekvenserna, som inte fanns representerande före omsamplingen, är satta till 0.



Figur 4.5. Förenklad beskrivning av interpolation, här illustrerad i fallet $w(x) = \Lambda(x)$ (linjär interpolation).

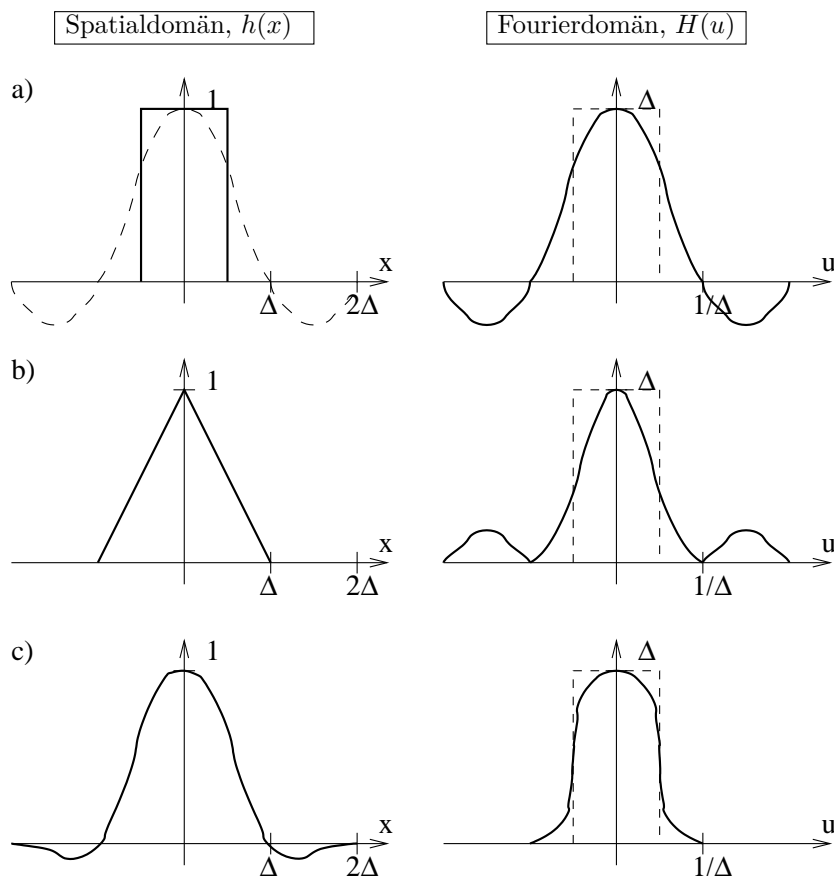


Figur 4.6. Ideal uppsampling med sinc-funktionen.

Vi ska nu jämföra några olika interpolationsmetodernas egenskaper genom att studera respektive interpolationsfunktions utseende i frekvensplanet.

Närmsta-granne metoden inses efter en stunds eftertanke vara identisk med faltning med en rektangelfunktion som sträcker sig över endast en pixel. I frekvensplanet motsvaras rektangelfunktionen av en sinc-funktion (Fig. 4.7 a). Likheten med det ideala filtret är knappast slående varför vi inte bör ställa så stora förväntningar på resultatet med denna metod. Metoden är dock snabb och kan lämpligen användas för translation av bilder och för zooming. Vid överföring till större matris är den dock oanvändbar eftersom vi endast får en kopiering av pixels.

Linjär interpolation är liktydigt med faltning med en triangeln (Fig. 4.7 b). Denna har i frekvensplanet betydligt mindre uttalade sidolober än föregående rektangelfunktion. Metoden ger hyggligt resultat, dock en smula oskarpt vilket indikeras av den avsmalnande toppen i triangelns fourierspektrum.



Figur 4.7. Interpolationsfunktioner $h(x)$ (vänstra kolumnen) och deras fouriertransform $H(u)$ (högra kolumnen). Den streckade funktionen är det ideala lågpasfilteret, $\text{sinc}(x/\Delta)$ i spatialdomänen respektive $\Pi(\Delta u)$ i fourierdomänen.

Eftersom ett idealt lågpasfilter i frekvensplanet motsvaras av en sinc-funktion i bildplanet kunde man förmoda att den ideala interpolationsfunktionen skulle vara just en sincfunktion. Sincen har dock tyvärr oändlig utsträckning med sidolober vars amplituder endast långsamt går mot noll. Någonstans måste den trunckas, troligen redan efter första sidoloben (för att inte få oacceptabelt lång beräkningstid) vilket resulterar i oönskade effekter. En naturlig tanke är att istället försöka hitta en funktion som approximerar sincen men som är 0 utanför ett visst intervall. Detta kan åstadkommas med cubic splines som är styckvis kontinuerliga funktioner sammansatta av tredjegradspolynom. Vi definierar en **cubic spline** $h(x)$ på följande sätt

$$h(x) = \begin{cases} a_3|x|^3 + a_2|x|^2 + a_1|x| + a_0, & |x| \in [0, 1], \\ b_3|x|^3 + b_2|x|^2 + b_1|x| + b_0, & |x| \in [1, 2], \\ 0, & \text{f.ö.} \end{cases} \quad (4.4)$$

Funktionen bör vara symmetrisk kring origo Dessutom ställer vi följande krav på $f(x)$:

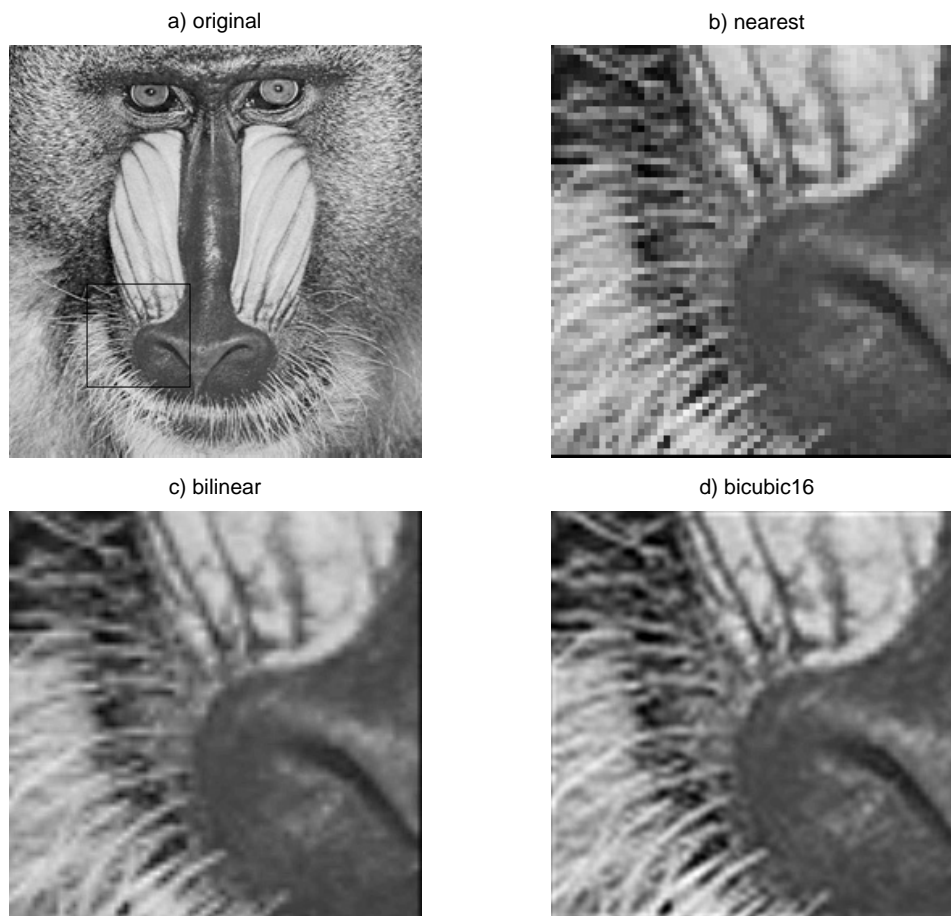
- Omsampling till samma matris skall ge en exakt reproduktion av originaldata.
Det innebär att: $h(0) = 1$, $h(1) = h(2) = 0$.
- Funktionen bör vara kontinuerlig i $x = 0$, $x = 1$ och $x = 2$.
- Derivatans bör vara 0 för $x = 0$ och $x = 2$ och dessutom kontinuerlig i $x = 1$.

Med hänsyn tagen till dessa krav blir $h(x)$ bestämd så när som på en konstant:

$$h(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & |x| \in [0, 1], \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & |x| \in [1, 2], \\ 0, & \text{f.ö.} \end{cases} \quad (4.5)$$

För negativa värden på konstanten a är $f(x)$ positiv i intervallet $[0, 1]$ och negativ i intervallet $[1, 2]$, dvs vi får något som liknar en trunkerad sinc-funktion. Det värde som i minsta kvadrat-mening ger en funktion så lik sinc-funktionen som möjligt är $a = -0.5$. Men andra värden är också möjliga. Det mänskliga ögat föredrar ofta lite skarpare kanter och då kan $a = -1$ vara ett alternativ. I fortsättningen kallar vi funktionen **4-point cubic spline** och ett ungefärligt utseende på denna visas i Fig. 4.7c). Vi får i frekvensplanet en god approximation av ett idealt lågpasfilter. Motsvarande funktion i 2D-fallet kallar vi **bicubic16**.

Fig. 4.8 illustrerar uppsampling, överföring av en bild till en större matris, här 4 gånger större. En del av originalbilden Fig. 4.8a) har förstörats genom närmsta granne interpolation i Fig. 4.8b), med linjär interpolation i Fig. 4.8c) och med bicubic16, ($a=-1$) i Fig. 4.8d). Bästa resultatet fås som väntat med den sistnämnda metoden. Linjär interpolation ger en mattare bild med suddigare kanter och taggigare linjer.



Figur 4.8. Uppsampling (förstoring). a) Originalbild (256×256) med ruta (64×64). b) Närmsta granne interpolation av rutan till storlek 256×256 . c) Linjär interpolation av rutan till storlek 256×256 . d) Bicubic16 interpolation ($a = -1$) av rutan till storlek 256×256 .

Vår beskrivning har hittills endast beskrivit en-dimensionell omsampling medan bilderna i Fig. 4.8 har bearbetats två-dimensionellt, vilket kräver en förklaring. Interpolationsfunktionen har applicerats två gånger efter varandra såsom illustreras i Fig. 4.9a,b). Först skapas en bild där den nya samplingstätheten uppträder i x -led. Dessa nya sampel användes sedan för en-dimensionell omsampling i y -led. Med en viss interpolationsfunktion h skapas slutresultatet $g(x, y)$ ur ingångsbilden $f(x, y)$ med

$$g(x, y) = f(x, y) * h(x)\delta(y) * h(y)\delta(x), \quad (4.6)$$

där den ideala två-dimensionella omsamplingen är dubbelsincen,

$$h(x)\delta(y) * h(y)\delta(x) = h(x) \cdot h(y) = \text{sinc}(x/\Delta) \cdot \text{sinc}(y/\Delta). \quad (4.7)$$

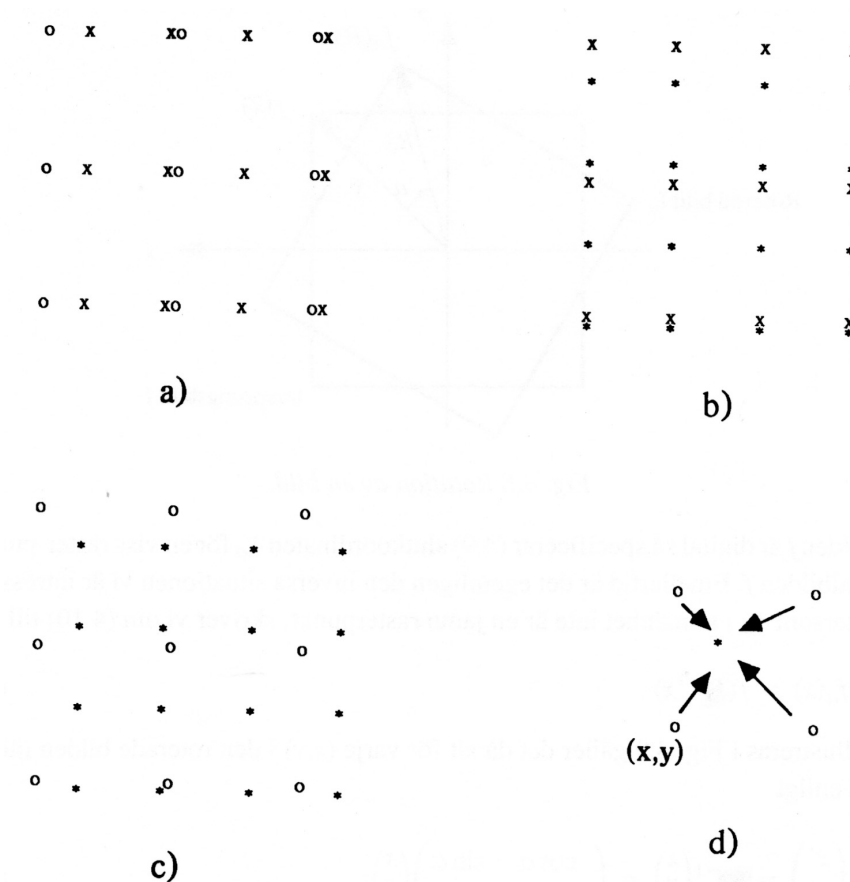
Approximationen av $h(x)$ och $h(y)$ med en triangelfunktion (linjär interpolering) innebär som antydes i Fig. 4.9d) att en viss utgångspixel interpoleras från sina fyra närmsta grannar. Låt den nya rasterpunktens koordinater vara (x', y') , vilka består av heltalsdelarna (x, y) och bråkdelarna (x_ϵ, y_ϵ) . Bilinjär interpolationen i två steg illustreras i Fig. 4.9a,b), vilket betyder att först sker interpolering i x -led, dvs

$$f_1(x', y) = (1 - x_\epsilon) \cdot f(x, y) + x_\epsilon \cdot f(x + 1, y), \quad x' = x + x_\epsilon, \quad (4.8)$$

där f är originalbilden vars rasterpunkter har markerats med \circ i Fig. 4.9a) och f_1 är mellanresultatsbilden vars rasterpunkter har markerats med \times i Fig. 4.9a,b). Därefter sker interpolation i y -led

$$f_2(x', y') = (1 - y_\epsilon)f_1(x', y) + y_\epsilon \cdot f_1(x', y + 1), \quad y' = y + y_\epsilon, \quad (4.9)$$

där f_2 är slutresultatsbilden vars rasterpunkter har markerats med $*$ i Fig. 4.9b).



Figur 4.9. Förstoring = tätare sampling. Originalbildens rasterpunkter har markerats med \circ , resultatbildens med $*$. a) En-dimensionell interpolation i x -led... b) ... följt av endimensionell interpolation i y -led... c),d) ... är ekvivalent med två-dimensionell interpolation.

Samma effekt kunde naturligtvis ha uppnåtts med en direkt två-dimensionell interpolering mellan de fyra närmsta sampelpunkterna i ingångsbilden dvs

$$f_2(x', y') = (1 - x_\epsilon)(1 - y_\epsilon) \cdot f(x, y) + x_\epsilon(1 - y_\epsilon) \cdot f(x + 1, y) \\ + (1 - x_\epsilon)y_\epsilon \cdot f(x, y + 1) + x_\epsilon y_\epsilon \cdot f(x + 1, y + 1), \quad (4.10)$$

vilket illustreras i Fig. 4.9c,d).

4.2 Rotation

Rotation av en bild f kring origo med vinkeln α enligt Fig. 4.10 ger en ny bild f_α där det roterade resultatet kan skrivas

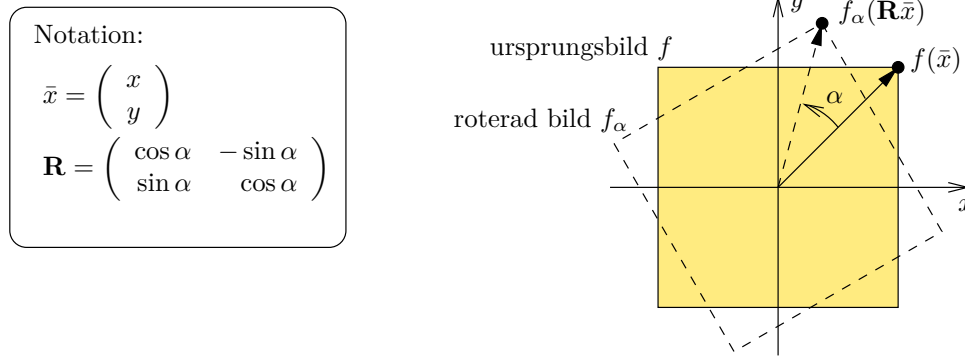
$$f_\alpha(x_\alpha, y_\alpha) = f(x, y), \quad (4.11)$$

där följande samband gäller mellan koordinaterna

$$\begin{pmatrix} x_\alpha \\ y_\alpha \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{R} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (4.12)$$

Låt oss införa beteckningarna $\bar{x}^T = (x, y)$ och $\bar{x}_\alpha^T = (x_\alpha, y_\alpha)$. Vi kan då sammanfatta (4.11) och (4.12) till

$$f_\alpha(\bar{x}_\alpha) = f_\alpha(\mathbf{R}\bar{x}) = f(\bar{x}). \quad (4.13)$$



Figur 4.10. Rotation av en bild.

Om bilden f är digital så specificerar (4.12) slutkoordinaten \bar{x}_α för en viss raster-punkt \bar{x} i originalbilden f . Emellertid är det egentligen den inversa situationen vi är intresserade av. Eftersom \bar{x}_α i allmänhet inte är en jämn rasterpunkt, skriver vi om (4.13) till

$$f_\alpha(\bar{x}) = f(\mathbf{R}^{-1}\bar{x}). \quad (4.14)$$

Som illustreras i Fig. 4.11 gäller det då att för varje (x, y) i den roterade bilden räkna ut (x', y') enligt

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{R}^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (4.15)$$

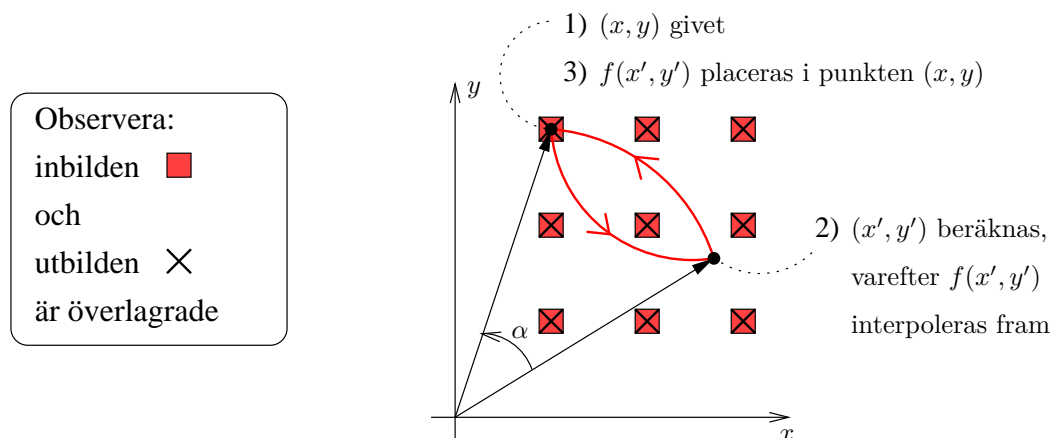
för att sedan med lämplig interpolation räkna fram $f(x', y')$ och utföra

$$f_\alpha(x, y) := f(x', y'). \quad (4.16)$$

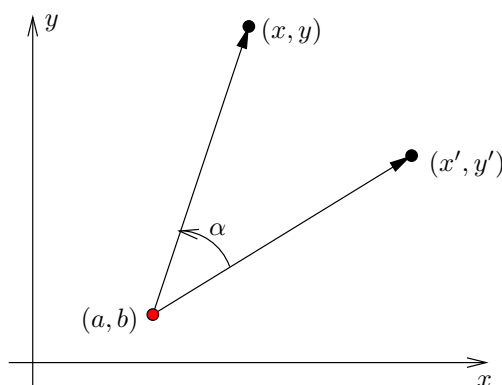
Punkten (x', y') är ju i det allmänna fallet ingen rasterpunkt utan ligger godtyckligt placerad. Först efter interpolation kan den egentliga rotationen utföras, dvs placera ett pixelvärde i punkten (x, y) . Observera alltså att rastret, samplingsmönstret, är gemensamt för f_α och f .

Rotation kring en punkt (a, b) är obetydligt mer komplicerad än rotation kring origo. Motsvarigheten till (4.15) blir

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x - a \\ y - b \end{pmatrix}, \quad (4.17)$$



Figur 4.11. Rotation av en digital bild innefattar beräkning av originalbildens värde i punkten (x', y') som skall föras till rasterpunkten (x, y) .



Figur 4.12. Rotation kring en godtycklig punkt (a, b) .

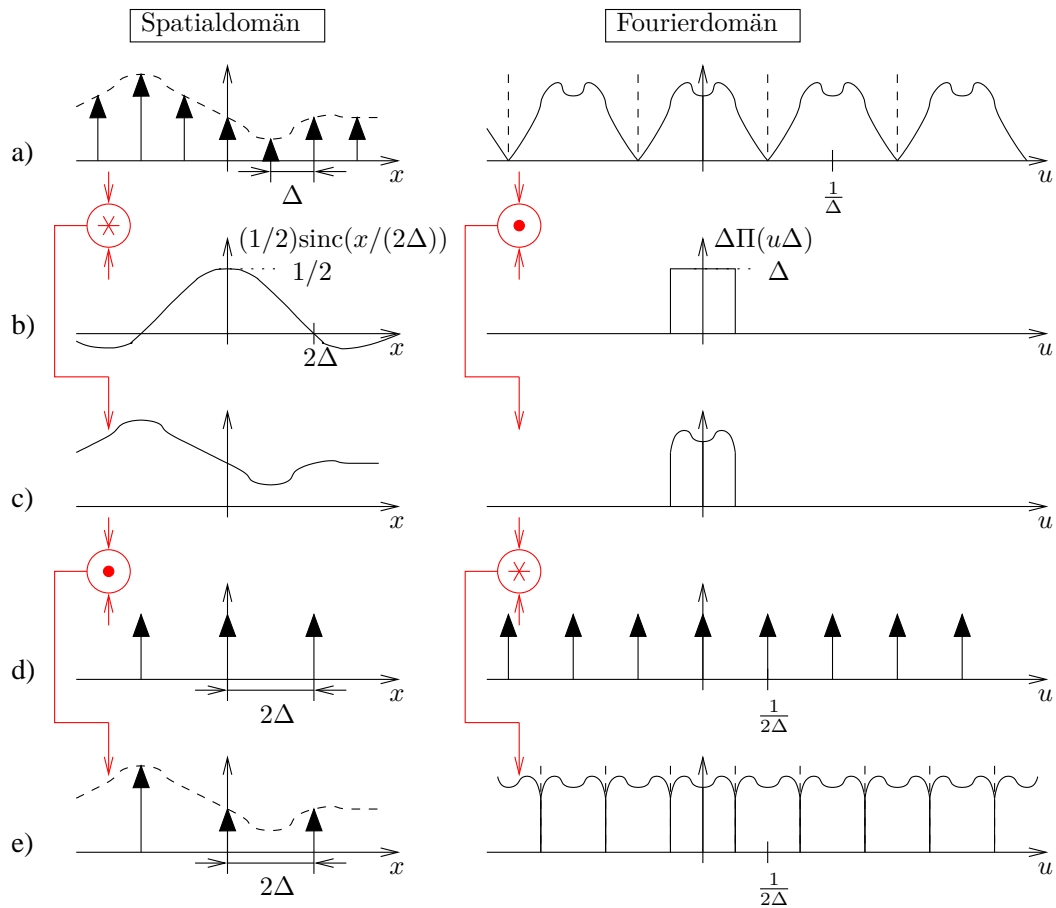
vilket illustreras av Fig. 4.12.

Som framgått av föregående avsnitt kan interpolation utföras mer eller mindre sofistikerat. Cubic spline (eller någon ännu mera sinc-liknande funktion) är ofta önskvärd. Vid ren omsampling utan rotation ligger de nya sampelpunkterna snyggt och prydligt i vertikala kolumner och horisontala rader i det gamla rastret som synes i Fig. 4.9. Detta gör det praktiskt möjligt att använda t ex en 10-punkters sinc-approximation i såväl x - som y -led $g(x, y)$, dvs totalt 20 multiplikationer-/ackumuleringar per pixelvärde i resultatet $g(x, y)$. Det totala omsamlingsförloppet är globalt separerbart.

4.3 Förminskning (nedsampling)

Metoderna i avsnitt 4.1 ovan, främst illustrerade av Fig. 4.2 - 4.7, kan naturligtvis i princip användas oförändrade även vid **nedsampling** (utglesning av sampelpunkterna = förminskning). Samplingsteoremet säger oss emellertid att det ökade samplingsavståndet innebär att vi inte kan härbärgera lika höga frekvenser som tidigare. Fig. 4.13 visar det en-dimensionella fallet för fallet med förminskning med faktorn $k = 2$. Om samplingsavståndet i den nya bilden är exakt 2 ggr originalets kan samtliga nya sampelpunkter fås att sammanfalla med en existerande; i det allmänna fallet faller de mellan två av originalets sampelpunkter. I fourierdomänen kan vi maximalt tillvarata vad som faller inom $-1/(k2\Delta) < u < 1/(k2\Delta)$ varför den rätta interpolationsfunktionen är $(1/k)\text{sinc}(x/(k\Delta))$. Som synes åtgår många sampel, närmare bestämt $2k$ st, redan för att approximera huvudloben i sinc-funktionen. Om vi skulle sträva att efterlikna även andra lobparet, som i fallet med cubic spline, Fig. 4.7 c), åtgår ytterligare $2k$ sampel. En två-dimensionell separabel cubic spline för omsampling / förminskning av en $N \times N$ pixel originalbild med faktorn k skulle därmed kräva

- $N \cdot 4k(N/k) = 4N^2$ MAC (multiplikation följt av ackumulering till tidigare summa) för N rader med vardera N/k nya sampelpunkter.



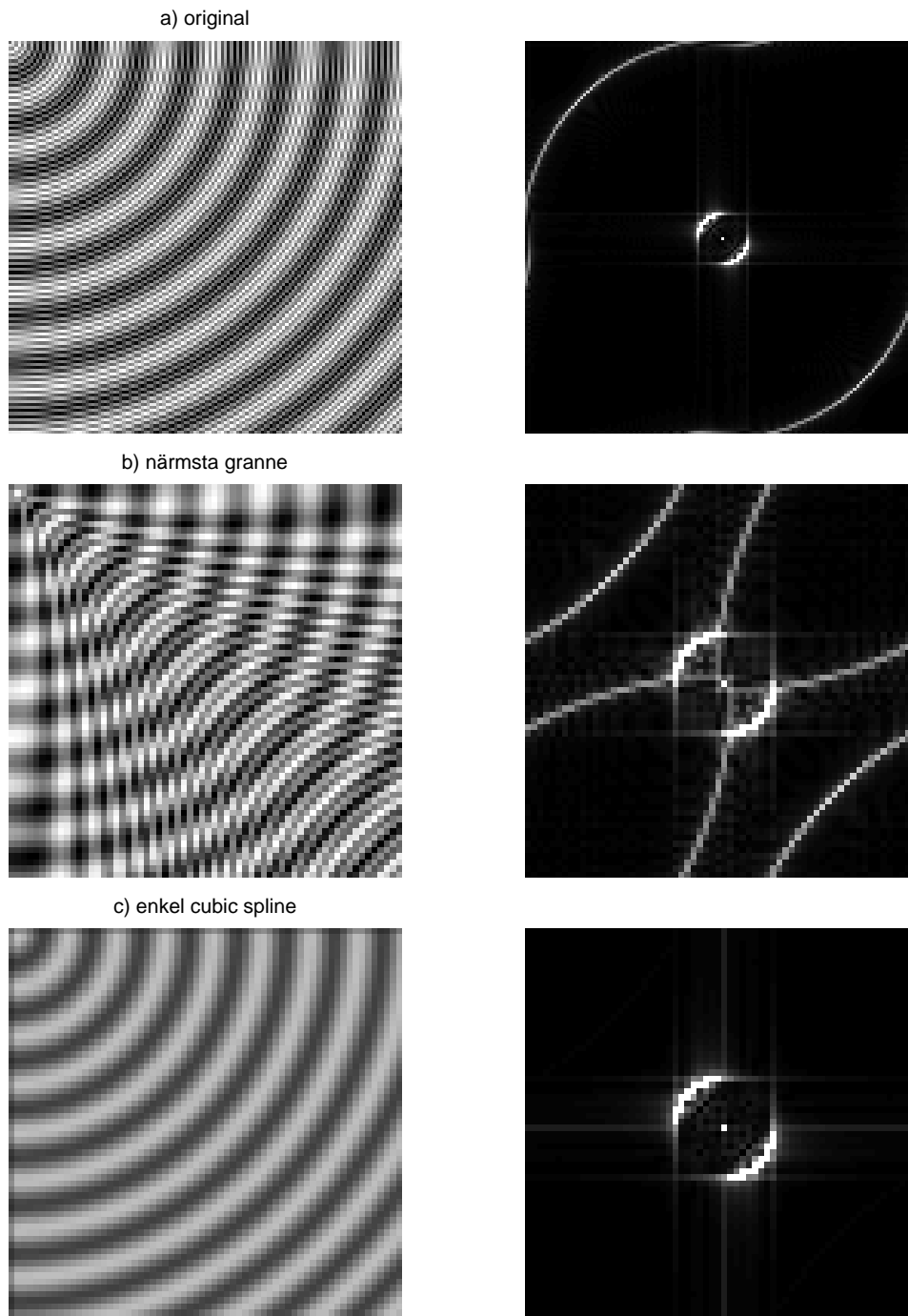
Figur 4.13. Nedsampling med faktorn 2 sker idealt med $\text{sinc}(x/(2\Delta))/2$.

- $(N/k) \cdot 4k(N/k) = 4N^2/k$ MAC för N/k kolumner med vardera N/k sampelpunkter.

Totalt åtgår alltså $4N^2(1 + 1/k)$ MAC.

En annan **enklare cubic spline** med samma bredd som huvudloben i sinc-funktionen definieras av

$$h(x) = \begin{cases} a_3|x|^3 + a_2|x|^2 + a_1|x| + a_0, & |x| \in [0, 1], \\ 0, & \text{f.ö.} \end{cases} \quad (4.18)$$



Figur 4.14. Förminskning med faktorn 2 av en testbild med 2 st radiella sinus-funktioner, en långsam och en snabb. Fouriertransform visas i högerkolumnen. a) Originalbild. b) Förminskning med närmsta granne. c) Förminskning med enkel cubic spline. För presentationen är resultatet zoomat till originalbildens storlek.

Liksom för funktionen (4.5) inför vi villkoren exakt reproduktion vid omsampling till samma matris, $h(0) = 1$, $h(1) = 0$, $h'(1) = 0$. Här gäller nu också $h'(0) = 0$. Parametrarna a_0 , a_1 , a_2 , a_3 blir då helt bestämda så att

$$h(x) = \begin{cases} 2|x|^3 - 3|x|^2 + 1, & |x| \in [0, 1], \\ 0, & \text{f.ö.} \end{cases} \quad (4.19)$$

Då denna funktion ska användas för nedsampling en faktor k används $(1/k)f(x/(k\Delta))$. Liksom funktionen (4.5) bevarar (4.19) DC-nivån perfekt men dess fourierspektrum, Fig. 4.13 b), är inte en lika god approximation av den ideala $\Delta\Pi(k\Delta u)$.

Den allra enklaste nedsamlingsmetoden är **närmsta granne**, $\Pi(x/\Delta)$. Den motsvaras i fourierdomänen av en alldeles för bred sinc-funktion $\Delta\text{sinc}(\Delta u)$ som ligger mycket långt från den rätta $\Pi(k\Delta u)$.

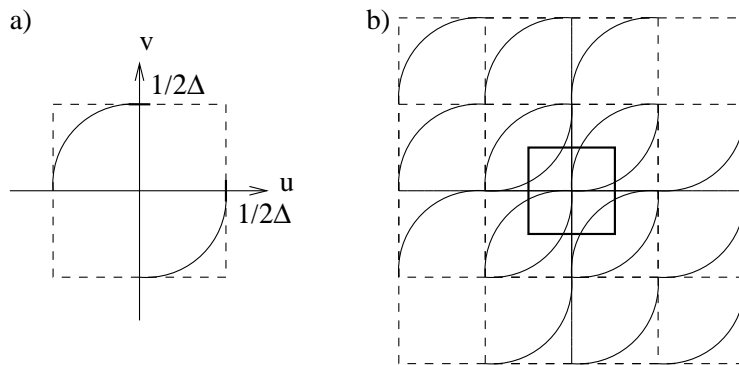
Om vi breddar beroendet av resultatpixeln till medelvärdet av en hel omgivning av k st ingångspixel ($k \times k$ i 2D), dvs $(1/k)\Pi(x/(k\Delta))$ motsvarar detta en smalare sinc-funktion i fourierdomänen, $\Delta\text{sinc}(k\Delta u)$, som är långt mer acceptabel. Denna metod kallar vi **närmsta grannar**. Metoden är betydligt snabbare än cubic spline. Cubic spline ger dock ett bättre resultat.

Ett slags mellanting mellan närmsta grann(ar) och cubic spline är **triangelfunktionen** som kommer i två varianter, dvs $\Lambda(x/(\Delta))$ och $(1/k)\Lambda(x/(k\Delta))$, där den första motsvarar enkel linjärinterpolation, medan den senare är ett bättre alternativ.

Fig. 4.14 visar förminskning av en testbild med faktorn 2 i såväl x-led som y-led under användning av tre olika interpolationsmetoder. Originalet är funktionen

$$\frac{1}{2} + \frac{1}{4} \sin\left(0.4\sqrt{x^2 + y^2}\right) + \frac{1}{4} \sin\left(3.2\sqrt{x^2 + y^2}\right) \text{ där } \Delta = 1,$$

dvs en DC-komponent som gör att gråskalan överallt är positiv, en radiellt formad sinusvåg av medel medelhög frekvens samt en andra radiellt formad sinusvåg med en frekvens vid själva Nyquistgränsen. För presentationen är resultaten zoomade i till originalets storlek. Den enklaste metoden är närmsta granne men den ger som synes mycket stor vinkningsdistorsion. Bilineär interpolation mellan de två närmsta grannarna i x-led och y-led ger likartad effekt. Om man däremot bildar varje utpixel med hjälp av en bred cubic spline funktion liknande sincens huvudlob i Fig. 4.13 b) undertryckes den höga frekvensen. Det speciella utseendet av invikningsfenomenet i Fig. 4.14 f) och h) förklaras av Fig. 4.15.



Figur 4.15. a) Skiss av fouriertransformen av testbilden i Fig. 4.14. b) Fouriertransformen upprepad med avståndet $1/2\Delta$ så att bidrag från alla 8 grannarna viks in i resultatet. Den nedsamlade bildens fouriertransform ges av den heldragna kvadraten med sidan $1/2\Delta$.

Litteraturförteckning

- [1] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. McGraw Hill, 1986.
- [2] Philip D. Cha and John I. Molinder. *Fundamentals of signals and systems*. Cambridge University Press, 2006.
- [3] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, Third Edition*. Prentice Hall, 2008.
- [4] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [5] Tim Morris. *Computer Vision and Image Processing*. Palgrave Macmillan, 2004.
- [6] Alan V. Oppenheim and Alan S. Willsky. *Signals and Systems*. Prentice Hall, 1997.
- [7] William K. Pratt. *Digital Image Processing*. Wiley Interscience, 1991.
- [8] John C. Russ. *The Image Processing Handbook, Third Edition*. CRC Press LLC, 1999.
- [9] Sune Söderkvist. *Tidskontinuerliga Signaler and system*. Tryckeriet E. Larsson AB, 1994.
- [10] Sune Söderkvist. *Från insignal till utsignal*. Tryckeriet E. Larsson AB, 2007.
- [11] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thompson Learning, 2008.
- [12] Anders Svärdröm. *Signaler och system*. Studentlitteratur, 1999.

