



NORGE

(12) **PATENT**

(19) NO

(11) **301860**

(13) B1

(51) Int Cl⁶ H 04 N 1/41, 7/26

Patentstyret

(21) Søknadsnr	920028	(86) Int. inng. dag og søknadsnummer	17.05.90, PCT/US90/02674
(22) Inng. dag	02.01.92	(85) Videreføringsdag	02.01.92
(24) Løpedag	17.05.90	(30) Prioritet	05.07.89, US, 375840
(41) Alm. tilgj.	05.03.92		
(45) Meddelt dato	15.12.97		

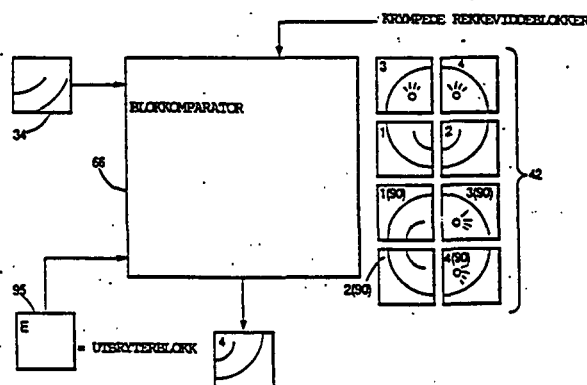
(73) Patenthaver	Michael F. Barnsley, 2153 Allaire Lane, Atlanta, GA 30345, US Alan D. Sloan, 1266 Holly Lane, Atlanta, GA 30329, US
(72) Oppfinner	Innehaverne
(74) Fullmektig	Onsagers Patentkontor AS, 0103 Oslo

(54) **Benevnelse** Fremgangsmåte og apparat til behandling av digitale data

(56) **Anførte publikasjoner** Ingen

(57) **Sammendrag**

Digitale bilde data blir behandlet automatisk ved å dele de lagrede bilde data i domeneblokker og rekkeviddeblokker. Rekkeviddeblokkene blir underkastet prosesser såsom en krympeprosess for å skaffe en avbildet blokk. Deretter blir for hver domeneblokk den avbildede rekkeviddeblokk som er mest lik domeneblokken bestemt og adressen til denne rekkeviddeblokk og prosessene som blokken ble underkastet blir kombinert til en identifikator som vedføyes en liste over identifikatorene for de andre domeneblokker. Listen av identifikatorer for alle domeneblokkene kalles en fraktal transformasjon og utgjør en komprimert representasjon av inngangsbildet. For å dekomprimere den fraktaltransformasjon og gjenvinne inngangsbildet, dannes et vilkårlig inngangsbilde som rekkeviddeblokker og rekkeviddeblokkene behandles på en måte spesifisert av identifikatorene for å danne en representasjon av det opprinnelige inngangsbilde.



Oppfinnelsen angår generelt bildebehandling og mer spesielt fremgangsmåter og apparatur som benytter kontraktive transformasjoner for representasjon av bilder og kompresjon av bildedata, idet fremgangsmåtene er angitt henholdsvis i innledningen av kravene 1, 23 og 24 og apparaturen i innledningen av kravene 31 og 34.

Det er en samling av viktige problemer som angår måten hvormed datamaskiner og andre digitale innretninger behandler bilder. Bredt sagt betegnes dette emne som "digital bildebehandling" og dekker analyse, lagring og overføring av bilder ved hjelp av digitale innretninger. Fire bestemte områder innenfor det brede emne digital bildebehandling og som er av særskilt interesse, er bildekompresjon, bilderepresentasjon, bildegjenkjenning og bildeanalyse.

Den foreliggende oppfinnelse kan anvendes på samtlige fire av disse områdene. Imidlertid er spørsmålene som vedrører hvert område omfattende og komplekse. Derfor vil bare to av områdene, bildekompresjon og bildegjenkjenning bli omtalt for å gi en bakgrunn for fordelene ved den foreliggende oppfinnelse.

For å forstå problemet med bildekompresjon, skal det betraktes et bilde, såsom et fotografi av en person eller en naturscene, representert som en matrise av 1024 x 1024 bildeelementer eller piksler. Pikslene kan være i svart/hvitt eller forskjellige gråtoner. Hvis det er 256 mulige gråtoner, så kan bildet være fullstendig beskrevet i digitale termer av en binær streng på 1024 x 1024 x 8 binære sifre (biter), hver med en verdi på 1 eller 0. En slik representasjon av et opprinnelig bilde er kjent som et "digitalisert" bilde, da det består av binære sifre og er egnet for behandling i en digital datamaskin. Bitene blir vanligvis plassert sammen i 8-bits grupper kjent som "bytes" og lagret sammen som en "datafil" for bruk i digitale innretninger såsom datamaskiner.

Hvis det ikke benyttes noen kompresjon, vil det være nødvendig med mer enn en million byter stort minne i en datamaskin for bare å lagre et enkelt bilde som beskrevet ovenfor. Bildekompresjon omfatter å redusere antallet byter som er nødvendig for å representere et bilde på en slik måte at det enten kan tas ut en eksakt gjengivelse av bildet (betegnet "tapsløs kompre-

sjon") eller en tilnærmet gjengivelse (betegnet "tapsbeheftet kompresjon") av bildet.

5 Det er fire hovedgrunner for å etterstrebe effektiv kompresjon av en datafil som representerer et bilde. For det første er overføringshastigheten for en datafil hovedsakelig proporsjonal med dens størrelse. Jo større filen er, desto lengre tid tar det å overføre den. For det annet er lagringsbehovet proporsjonalt med filstørrelsen. For å lagre filer som representerer seksti ukomprimerte bilder av den art som er nevnt ovenfor, kreves det 60 megabyter med lagringsplass. Hvis representasjonen av bildet kan komprimeres med 50 %, kan det lagres dobbelt så mange bilder i den samme lagringsstørrelse. For det tredje koster komprimerte bilder mindre å overføre, behandle og arkivere og gjenfinne enn ukomprimerte bilder. For det fjerde kan komprimerte bilder behandles hurtigere enn ukomprimerte bilder. Anta f.eks. at et prøvebilde sammenlignes mot et bibliotek av lagrede bilder for å bestemme hvorvidt en gjerdesmutt forekommer i prøvebildet. Selv om den eneste forskjellen mellom bildene kan være en liten fugl, måtte filene som representerer bildene, for å detektere denne forskjellen med bruk av ukomprimerte data sammenlignes bit for bit, piksel for piksel. Hvis bildene var riktig komprimert, kunne sammenligningen utføres raskere. Ene og alene av den fjerde grunn vil datakompresjon høyst sannsynlig spille en betydelig rolle i kunnskapsbaserte databehandlingssystemer.

Eksempler på nåværende behov for effektiv kompresjon er som følger:

- 25
- (1) Overføring av medisinske bilder over telefon. Dette innbefatter medisinske røntgenbilder og CAT-skann med høy oppløsning. Medisinsk rådgivning kunne fås fra en rekke konsulenter på fjerne steder, gitt at overføringskostnadene er tilstrekkelig lave og overførings tidene er tilstrekkelig korte.

30

 - (2) Overføring av videodata fra datamaskin til datamaskin over telefon. I mange situasjoner må fjern datamaskiner forbindes med telefonlinjer og bruk av et modem. Kommunikasjonen mellom datamaskinene begrenses av "båndbredden" til linjen mellom datamaskinen og hastigheten til modemmet. For standard telefonlinjer er de høyeste hastigheter hvormed datamaskiner kan overføre data til hverandre

35

9600 biter/s. Hvis et enkelt videobilde oppløses i 128 x 128 x 8 eller 131 076 biter, vil video informasjonen ved 30 bilder/s genereres med 3 932 160 biter/s. Følgelig er det nødvendig med et kompresjonsforhold på 409:1 for denne anvendelse.

5

- (3) 64 Kbit/s telefonlinjer for videooverføring. Målet er å være i stand til å tillate forbrukere å få adgang til filmer på forespørsel. Kvaliteten som kreves av videoen, er høy. Følgelig er det nødvendig med en god kompresjonsmetode.

10

- (4) Høydefinisjonsfjernsyn (HDTV) og avansert fjernsyn (ATV). Det er sannsynlig at De Forente Stater vil kreve at enhver form av HDTV må være kompatibel med NTSC-signaler, de standard kringkastede fjern synssignaler som for tiden overføres i USA. HDTV vil gi bilder med omtrent den dobbelte horisontale og vertikale oppløsning i forhold til NTSC-fjernsynsbilder. For å oppnå dette er det foreslått at ekstra båndbredde skal benyttes av kringkastingsstasjonene. Slik ekstra båndbredde ville ikke være nødvendig hvis effektive midler for bildekompresjon stod til rådighet og de ekstra data som var nødvendig for HDTV, kunne overføres via eksisterende kommunikasjonsbånd.

15

20

- (5) Anvendelser av typen "Digital Video Interaction" (DVI). Målet er å lagre så mye videodata som mulig på et medium såsom optisk plate. Jo høyere kompresjon som kan oppnås, dess flere anvendelser av interaktiv gjenfinning ved videorater vil være praktisk gjennomførlige.

25

- (6) Mer effektiv fax- (faksimile-) overføring. Nåværende fax-maskiner overfører en tekstsider på ca. 15 s. Selv om en viss kompresjon av svart-hvitt-bilder kan oppnås med fax-maskiner av type 3, ville en økning i kompresjonsforholdet med en faktor ti tillate tekstoverføring via fax med en rate på 40 sider/min. Dette vil gi betydelige økonomiske besparelser.

30

35

- (7) Mer effektiv datagjenfinning. Etter hvert som datamengden som lagres i en datamaskin blir større, øker aksesstiden for å gjenfinne

dataene på grunn av den økende størrelse på datafilene. Bildekompresjon kan benyttes for å oppnå hurtigere aksess til dataene ved å redusere mengden av lagerplass som må søkes for å gjenfinne den.

5

Kjente fremgangsmåter og kjent utstyr for eksakt kompresjon oppnår forhold på mellom 0,5 og 0,1. Det vil si at antallet biter som er nødvendig for å lagre og reprodusere en eksakt kopi av originalbildet er mellom halvparten og en tiendedel av det opprinnelige antall biter. En slik ytelse er tydeligvis ikke tilstrekkelig til å oppfylle kravene til noen av de nåværende anvendelser som er anført ovenfor.

10

Dessuten er en del kjente digitale datakompresjonsmetoder beregningsmessig komplekse, selv om de er egnet for visse anvendelser, og kan således behøve flere timer med manuell betjening av en datamaskin for å oppnå komprimerte data.

15

Ytterligere andre kjente datakompresjonsmetoder benytter oppslagstabeller som del av kompresjonsprosedyren. Prosedyrer som benytter oppslagstabeller kan være anvendelige i visse sammenhenger, så som hvor klassen av bilder som komprimeres er liten. Imidlertid kan bruken av oppslagstabeller begrense evnen til å skaffe adaptive muligheter som kan benyttes i anvendelser så som alminnelig anvendbar videobehandling, herunder et stort antall forskjellige bilder og andre anvendelser hvor bildeklassen ikke kan begrenses på forhånd. Når klassen av bilder som skal komprimeres er stor, må også oppslagstabellene være store og vil kreve bruk av ekstremt store mengder systemhukommelse. Videre reduseres kompresjonsforholdene etter hvert som størrelsen på oppslagstabellene økes.

20

25

Følgelig er det nødvendig med forbedrede metoder og forbedret utstyr for kompresjon av billedata.

30

Et annet viktig problem i digital bildebehandling angår gjengivelsen av bildet i en datamaskin. Virkelige bilder av verden slik vi ser dem, er ikke digitale, dvs. diskrete. I stedet har disse bildene en kontinuerlig utstrakt natur. Et bilde for bevissthetens øye ses som om det lot seg forstørre i det uendelige. Hvis vi f.eks. tenker på et lite bilde som en liten sort katt, begynner vi med begrepet

35

"sort katt" og ser i bevisstheten en liten svart, formet silhuett. Hvis vi tenker oss at bildet forstørres kontinuerlig, fremkommer ytterligere detaljer i bildet i bevisstheten, såsom en teksturert pels og værhår. Vi er i vår bevissthet i stand til kontinuerlig å forstørre bildet for å fremskaffe ethvert nødvendig detalj-nivå.

Datamaskiner har på den annen side den evnen bare i den mest rudimentære grad. Det er for tiden f.eks. mulig å definere bare enkle bilder, såsom en linje, i en datamaskin slik at bildet beholder et høyt detaljnivå ved hver forstørrelse. Dette skjer ved å lagre en definisjon i datamaskinen av f.eks. en linje som spesifiserer det helt essensielle ved linjen uten å spesifisere dens kvaliteter ved et eller annet bestemt forstørrelsesnivå. Linjen defineres i datamaskinen som begrepet for en geometrisk størrelse som strekker seg mellom to bestemte punkter. Når det er ønskelig, benyttes denne definisjonen til å frembringe et bilde med hvilken som helst ønsket grad av forstørrelse.

Hvis skjermen hvorpå et bilde av den lagrede definisjon av en linje skal vises, har en oppløsningsevne på f.eks. tre hundre punkter pr. tomme og linjebildet ønskes vist som et bilde med en lengde på to tommer, vises et bilde som består av et visningsmønster med en lengde på seks hundre punkter og f.eks. en bredde på tre punkter. Hvis et større bilde er ønsket vist fra den samme lagrede definisjon av en linje, kan skjermen vise et bilde som er tre punkter bredt og tolv hundre punkter langt. Det ville være ønskelig å skaffe en fremgangsmåte til å representere komplekse bilder, så som en svart katt, i en datamaskin med den samme art av skala uavhengig oppløsning.

Det er således et behov for en måte til å representere bilder i en datamaskin og som, selv om de legges inn og gis ut med bruk av digitale innretninger og lagres på digital form, i realiteten er kodet på en måte som tillater bildet å håndteres som om det var kontinuerlig. Løsningen på dette problem ville bety at datamaskiner behandler bilder på en måte som er mer analog med den måte hvorved den menneskelige bevissthet lagrer og analyserer bilder. Denne løsningen antas å være sentral for mange viktige problemer som vedrører bildegjenkjenning og -analyse i digitale beregnings innretninger.

Anvendelser av bilderepresentasjon innbefatter:

- (1) Gjenkjennelse av hva som er et bilde.
- (2) Mønstergjenkjenningsanvendelser.
- (3) Sjekklesemaskiner. Et bilderepresentasjonssystem ville tillate
5 gjenkjennelse av signaturer og ut visking av overflødig
bakgrunnsmateriale.
- (4) Maskiner for å detektere menneskelagede objekter i et bilde.
- (5) Bildeanalysatorer anbragt i satellitter i jordbane for å påvise skog
branner og overvåke avskogingsaktiviteter.
- (6) Passive transpondere til bruk i supermarkeder.
- 10 (7) Systemer for å motta bilder med en oppløsning og gi dem ut med
høyere oppløsning, såsom strektegninger, fotografier og bilde for-
sterkingssystemer.
- (8) Automatiske innretninger for å gjenkjenne folks ansikter. Disse kan
benyttes i sikkerhets anvendelser.
- 15 (9) Høydefinisjonsfjernsyn. Bestemte anvendelser omfatter bilde for-
sterkning, bruk av det vertikale slukkeintervall til å skaffe ytterli-
gere informasjon og overføring av komprimerte bilderepresenta-
sjonsdata for høydefinisjonsbilder.
- (10) Analyse av bilder av planterøtter med det formål å skaffe vi-
20 tenskapelige data vedrørende roten.

Det er derfor ønskelig å skaffe et effektivt middel for bildegjengivelse og
som tillater digital gjengivelse som er oppløsningsuavhengig og som ikke
iboende beskriver diskrete bilder, men i stedet analoge (kontinuerlige) bilder.
25 Representasjonsmetoden må kunne anvendes like godt på digitale som på
analoge bilder.

Det er likeledes ønskelig å utvikle bildekompresjonsmetoder og -utstyr hvis
inngangssignaler er sett av data, svarende til en todimensjonal matrise av
30 numeriske attributter for et digitalisert bilde og hvis utgangssignaler er et
mindre datasett hvorfra originalen kan regenereres eksakt eller tilnærmet. Det
er videre ønskelig å skaffe forbedrede bildekompresjonsmetoder og -utstyr
som kan benyttes til eksakt eller ikke-eksakt bildegjengivelse og være i stand
til å forbedre kompresjonsforholdet med inntil flere større lsesordner. Slike
35 fremgangsmåter og slikt utstyr burde frembringe digitale data som svarer til
et gitt bilde og burde anvende de digitale data for bildekompresjon, kommu-
nikasjon, analyse eller lagring og påfølgende regenerering. Dessuten burde

slike fremgangsmåter og slikt utstyr være i stand til å representere farge- og intensitetsinformasjon som en ytterligere dimensjon av bildet og tillate kompresjon og dekompresjon av denne fargeinformasjonen.

- 5 Det er også ønskelig å skaffe metoder og utstyr til automatisk bestemmelse av parametre som utgjør et komprimert datasett som representerer et originalbilde og er i stand til å frembringe en tilnærmet gjengivelse av originalbildet ved å benytte det komprimerte datasett på et sett av data som representerer et vilkårlig utgangsbilde.

10

Det er også ønskelig å skaffe fremgangsmåter og utstyr til å komprimere og dekomprimere digitale data som gjengir film bilder med bruk av den fraktale transformasjon.

- 15 Endelig er det ønskelig å skaffe fremgangsmåter og utstyr til å frembringe bilder som etter flere gjennomganger gjennom en fullstendig kodings- og dekodingsprosess kommer ut som om de bare hadde gått gjennom prosessen en gang og som gir et middel til å ekspandere bildene til høyere oppløsning.

- 20 Kort beskrevet skaffer den foreliggende oppfinnelse forbedrede fremgangsmåter og utstyr til å representere bildedata på komprimert form på en selvrefererende måte. De komprimerte data dannes i en digital datamaskin ved en prosess av gjentatte sammenligninger mellom bildedata som representerer små biter av bildet, kjent som domeneblokker og krympede rekkeviddeblokker og registrerer adressene for optimale tilpasninger. Bildet dekodes ved en invers prosess.

- 30 De ovennevnte hensikter og fordeler oppnås i henhold til oppfinnelsen med fremgangsmåter som er kjennetegnet ved trekk som fremgår av karakteristikken til henholdsvis de selvstendige krav 1, 23 og 24 samt med et apparat som er kjennetegnet ved trekk som fremgår av karakteristikken til de selvstendige krav 31 og 34.

- 35 Den foretrukne utførelse av den foreliggende oppfinnelse benytter matematiske operasjoner kjent som lokale affine transformasjoner. Disse transformasjonene benyttes i prosessen for koding av digitale data som representerer et bilde. De kodede utgangsdata utgjør en "fraktaltransformasjon" av et bilde

og består av koeffisienter for de affine transformasjoner. Forskjellige fraktaltransformasjoner svarer til forskjellige bilder.

De fraktale transformasjoner behandles iterativt i dekoding operasjonen. I motsetning til hva som intuitivt kan forventes er det blitt oppdaget at hvis

5 kodings- og dekodings prosedyrene i henhold til oppfinnelsen følges, dannes det opprinnelige bilde eller et bilde som ser ut som original bildet, etter et endelig antall iterasjoner.

10 I henhold til formålet og prinsippene for oppfinnelsen skaffes en fremgangsmåte til automatisk å behandle digitale billedata bestående av bildeinformasjon. Fremgangsmåten omfatter trinn utført i en dataprosessor for å lagre billedata i dataprosessoren og å generere en rekke entydig adreserbare domeneblokker fra de lagrede billedata, idet hver av domeneblokkene

15 representerer en forskjellig del av bildeinformasjon slik at all den lagrede bildeinformasjon befinner seg i minst en av domeneblokkene. Fremgangsmåten omfatter ytterligere trinn for fra de lagrede billedata å danne en rekke entydig adresserbare avbildete rekkeviddeblokker svarende til forskjellige undermengder av billedata, hver av undermengdene med en entydig adresse,

20 idet dannelsenstrinnet innbefatter et undertrinn for for hver av de avbildete rekkeviddeblokker å eksekvere en tilsvarende prosedyre på en av undermengdene av billedata som svarer til den avbildete rekkeviddeblokk. Fremgangsmåten omfatter videre trinn for å tilordne til den avbildete rekkeviddeblokken entydige identifikatorer som hver spesifiserer både den

25 tilsvarende prosedyre og adressen til den tilsvarende undermengde av billedata, idet de avbildete rekkeviddeblokker som mest nøyaktig tilsvare domeneblokkene, velges i henhold til forhåndsbestemte kriterier, og å representere bildeinformasjon som en mengde av entydige identifikatorer svarende til de valgte avbildete rekkeviddeblokker.

30 I henhold til et annet trekk skaffer oppfinnelsen en fremgangsmåte til å generere et bilde ved hjelp av en dataprosessor fra en mengde av identifikatorer, idet dataprosessoren innbefatter et minne med en rekke buffere som hver har en rekke adresserbare områder spesifisert av bufferadresse, idet hver

35 av identifikatorene svarer til et adresserbart område i bufferne og innbefatter både en prosedyre og en bufferadresse. Fremgangsmåten omfatter trinn utført av en dataprosessor for å lagre et forhåndsbestemt mønster av digitale data i

en av bufferne betegnet som en kildebuffer, og for identifikatorene å bestemme mønstre av data som svarer til hver identifikator ved å benytte prosedyren i denne identifikator på den del av kildebufferen som angis av bufferadressen i denne identifikatoren, å lagre i hver av de adresserbare områder i en annen av bufferne, betegnet som en destinasjonsbuffer, det tilsvarende mønster av data for identifikatoren svarende til denne adresserbare del, å gjenta de foregående to trinn med destinasjonsbufferen betraktet som kildebufferen inntil forhåndsbestemte kriterier oppfylles, og å levere innholdet av destinasjonsbufferen som data representative for et bilde når de forhåndsbestemte kriterier blir oppfylt.

Den ledsagende tegning som er innbefattet i og utgjør en del av denne beskrivelse, viser utførelser av oppfinnelsen og tjener sammen med beskrivelsen til å forklare prinsippene for den foreliggende oppfinnelse.

Fig. 1 viser et digitalisert svart-hvitt bilde og prosessen med å dele bildet opp i domeneblokker.

Fig. 2 viser prosessen for å dele bildet på fig. 1 i rekkeviddeblokker.

Fig. 3 viser prosessen for en forskjøvet oppspalting av bildet på fig. 1 med det formål å danne ytterligere rekke viddeblokker.

Fig. 4 viser prosessen for å danne krympede rekkeviddeblokker fra rekkeviddeblokkene på fig. 2 og 3.

Fig. 5 viser logisk flytkart for fremgangsmåten til automatisk behandling av digitale bilde data og som utgjør en foretrukket utførelse av oppfinnelsen.

Fig. 6 viser et diagram av en dataprosessor for utførelse av fremgangsmåten vist på fig. 5.

Fig. 7 og 8 viser mer detaljert prosessen for å generere krympede rekkeviddeblokker.

Fig. 9 viser prosessen for å velge den krympede rekkeviddeblokk som mest nøyaktig motsvarer en domeneblokk.

Fig. 10 viser prosessen for å frembringe en hausdorffavstand.

Fig. 11 viser en fraktal transformasjon av bildet på fig. 1.

5

Fig. 12 viser et logisk flytkart som gjengir en metode for automatisk å frembringe en fraktaltransformasjon som representerer svart-hvitt bildedata.

10

Fig. 13 viser et logisk flytkart for en metode til automatisk å generere et bilde med en dataprosessor fra et sett av identifikatorer bestående av fraktaltransformasjonen frembragt ved metoden på fig. 12.

15

Fig. 14 viser et blokkdiagram av en foretrukket utførelse av utstyr for de-koding av en fraktaltransformasjon.

Fig. 15 viser et kompresjonssystem som benyttes til å dekode den fraktale transformasjon på fig. 11.

Fig. 16 viser prosessen med å benytte den fraktale transformasjon på fig. 11 på et bilde i buffer A på fig. 15 for å skaffe bilde i en buffer B.

Fig. 17 viser et ytterligere trinn av prosessen på fig. 16.

Fig. 18 viser dannelsen av et utgangsbilde fra dekompresjonssystemet på fig. 15.

Fig. 19 viser et logisk flytkart av en foretrukket utførelse av metoden på fig. 13.

Fig. 20 viser et logisk flytkart for et alternativ til metoden gjengitt på fig. 19, med bruk av bare en enkelt bildebuffer.

Fig. 21-24 viser frembringelsen av en fraktaltransformasjon som representer et gråskalabilde.

35

Fig. 25 viser en metode for å representere et gråskalabilde med fire mulige gråverdier på binær form.

Fig. 26 viser en fraktaltransformasjon av gråskalabildet på fig. 21.

5 Fig. 27 viser et logisk flytdiagram som gjengir en metode for automatisk å frembringe en fraktaltransformasjon som representerer gråskalabildedata.

Fig. 28 viser et dekompresjonssystem for dekoding av den fraktale transformasjon vist på fig. 26, for å skaffe det opprinnelige gråskalabilde på fig. 21.

10 Fig. 29 viser et logisk flytkart for en metode for å dekode den fraktale gråskalatransformasjon på fig. 26.

Fig. 30 viser et logisk flytkart for en metode for komprimering og dekomprimering av fargebilder.

15

Fig. 31 viser et blokkdiagram av et apparat som utgjør en foretrukket utførelse for gjennomføring av metoden på fig. 20.

20 Fig. 32 viser et flytkart av en fremgangsmåte til å generere komprimerte digitale data som representerer bildeinformasjon i et filmbilde.

Fig. 33 viser et logisk flytkart av en fremgangsmåte til automatisk generering av digitale data som representerer bilder i en film.

25 Fig. 34 viser et blokkdiagram av et videofonapparat.

Før den foretrukne utførelse av oppfinnelsen beskrives, vil det gis bakgrunnsinformasjon og omtale vedrørende arten av oppfinnelsen i relasjon til både dekompresjon og bilderepresentasjon.

30

Mange bilder inneholder overflødig informasjon. F.eks. kan et bilde av en naturscene inneholde en eng, trær og himmel. Straks man vet hvordan et bestemt gresstrå ser ut, har man nyttig informasjon om andre strå. Således has det enorme mengder overflødig informasjon i et bilde av en eng. Furu
35 nålene på et furutre er på ingen måte slumpmessig plassert i bildet, de er organisert av treets underliggende struktur. Strukturene til skyene styres med god tilnærmelse av kort uttrykte ligninger i fysikken, såsom navier-stokes

ligningene. Disse eksempler indikerer at signifikant kompresjon kan oppnås med data som representerer bilder av f.eks. enger, trær og himmel.

5 Således er overflødigheit eller redundans en vesentlig faktor i bildekompre-
sjon og bilderepresentasjon av bilder så som naturscener, portretter og
datagrafikkbilder som kan uttrykkes repetitivt etter prosesser med affin
transformasjon og kontraksjon. Som tidligere nevnt, er en furunål i et bilde
av en skog mer eller mindre lik enhver annen furunål i dette bilde. Et
10 komprimert bilde behøver ikke beskrivelse av hver furunål i dette bilde, i
stedet blir en enkelt furunål beskrevet og deretter blir dens basisinformasjon
benyttet via matematiske algoritmer for å gjengi andre furunåler i bildet på
en naturlig måte. Dette kan sammenlignes med å beskrive bokstav i en
tekstside og deretter beskrive andre forekomster av denne bokstaven uttrykt
ved den første forekomst.

15 Det er essensielt to trinn i digital bildebehandling. Den første berører
transformasjonen av bildet til en annen representasjon. Dette inkluderer ofte
datakompresjon, men ikke nødvendigvis. Den annen berører gjenskaping av
ori ginalbildet, hvilket kan omfatte dekompresjon.

20 Med henvisning til tegningen hvor de samme elementer er angitt med de
samme henvisningsbetegnelser, vil foretrukne utførelser av fremgangsmåtene
og apparatet i henhold til den foreliggende oppfinnelse nå beskrives i detalj.
Først skal en datakompresjonsmetode og -apparat i samsvar med oppfinnel-
25 sen beskrives.

Fig. 1 viser et digitalisert svart-hvitt bilde 30 dannet av en rekke piksler 32.
Bildet lagres som bildedata i en dataprosessor, og en bit representerer
bildeverdien (svart eller hvitt) for hvert piksel. Den foretrukne utførelse av
30 oppfinnelsen benytter en rekke entydig adresserte blokker 34, kalt domene-
blokker, som genereres av de lagrede bildedata. Hver av domeneblokkene
representerer en forskjellig del av bildeinformasjonen, slik at alle domene-
blokkene til sammen inneholder all den lagrede bildeinformasjon.

35 De lagrede bildedata blir også benyttet i den foretrukne utførelse til å danne
en rekke entydig adresserbart av bildete rekkeviddeblokker. Avbildete
rekkeviddeblokker svarer til en forskjellig undermengde av bildedata, idet

hver av undermengdene har en entydig adresse med hensyn til billedataene. Fortrinnsvis har rekkeviddeblokkene et større antall piksler enn domeneblokkene og frembringes ved for hver av de avbildete rekkeviddeblokker å eksekvere en tilsvarende prosedyre på undermengden av billedataene som svarer til denne avbildete rekkeviddeblokk. Prosedyren kan innbefatte slike prosesser som krymping, rotasjon, oppdeling og invertering. I en foretrukket utførelse brytes f.eks. bildet 30 opp i rekkeviddeblokken 36 som vist på fig. 2, og inverteres for å frembringe ekstra rekkeviddeblokker 38. Ytterligere rekkeviddeblokker 40 kan frembringes ved å velge andre undermengder av det innmatede digitaliserte bilde 30, som vist på fig. 3.

En krympeprosess, vist på fig. 4, blir fortrinnsvis benyttet på rekkeviddeblokkene 36,38 og 40 for å frembringe krympede rekkeviddeblokker 42. Krympeprosessen sikrer at den krympede rekkeviddeblokk 42 inneholder samme antall piksler som domeneblokkene 34.

Begrepene domeneblokker og rekkeviddeblokker vil være nyttige for forståelsen av prinsippene for den foreliggende oppfinnelse som i en henseende skaffer en fremgangsmåte, utført i en dataprosessor, for å behandle digitale billedata bestående av bildeinformasjon, som vist på fig. 5. I trinn 42 lagres først billedata i en dataprosessor. Trinn 44 på fig. 5 omfatter å generere en rekke entydig adresserbare domeneblokker fra de lagrede billedata, idet hver av domene blokkene representerer en forskjellig del av bildeinformasjonen slik at den lagrede bildeinformasjon rommes i minst en av domeneblokkene. Trinn 46 omfatter fra de lagrede data å frembringe en rekke entydig adresserbare avbildete rekkeviddeblokker svarende til de forskjellige undermengder av billedata, hver av undermengdene med en entydig adresse, idet dannelsenstrinnet fortrinnsvis innbefatter for hver av de avbildete rekkeviddeblokker et undertrinn for å eksekvere en tilsvarende prosedyre på en av undermengdene av billedataene som svarer til den avbildete rekkeviddeblokk. Trinn 48 omfatter å tilordne entydige identifikatorer til de respektive tilsvarende avbildete rekkeviddeblokker, idet hver av identifikatorene for den tilsvarende rekkeviddeblokk spesifiserer en prosedyre og en adresse for den tilsvarende mengde av billedata. Dernest omfatter trinn 50 for hver av domeneblokkene å velge den av de avbildete rekkeviddeblokker som mest nøyaktig viser overenstemmelse i henhold til for

håndsbestemte kriterier. Endelig omfatter trinn 52 å gjengi bildeinformasjonen som en mengde av valgte identifikatorer.

- 5 Et blokkdiagram for en passende dataprosessor 60 er vist på fig. 6. Dataprosessoren 60 innbefatter en inngangsinnetning 62, såsom en bildeskanner. Inngangsinnetningen 62 er forbundet med en sentralprosessor 64. Sentralprosessoren (CPU) 64 omfatter en blokkomparator 66, kontrollogikk 68 og en avbildningsprosessor 70. CPU 64 er forbundet til en utgangsinnetning 72 såsom en datasender eller en platestasjon og til en minneinnetning 74.
- 10 Bildedataene lagres i minne innetningen 74. Minneinnetningen 74 innbefatter par av buffere 76,78. Buffer 76 benyttes til å lagre en rekke domeneblokker 34 og buffer 78 benyttes til å lagre rekke viddeblokker 36,38,40 og krympede rekkeviddeblokker 42.
- 15 Kontrollogikken 68 i CPU 64 eksekverer programmer lagret i minnet 74 for henholdsvis å initiere blokksammenligning og avbildningsfunksjoner i avbildningsprosessoren 70.

- I den foretrukne utførelse blir bildedata mottatt fra inngangsinnetningen 62, av kontrollogikken 68 omformet til en matrise av digitaldata med forhåndsbestemt størrelse, som f.eks. 256 x 256 piksler og som svarende til trinn 42 på fig. 5, lagres i minnet 74. I henhold til trinn 44 blir matrisen deretter omdannet til en matrise av domeneblokker 34 som hver f.eks. innbefatter 16 piksler anordnet i fire rader og fire spalter.

- 25 I henhold til trinn 46 blir de lagrede bildedata dannet som rekkeviddeblokker 36 som inneholder et større antall piksler enn antallet piksler i domeneblokkene 34. Disse rekkeviddeblokkene blir deretter gitt entydige adresser med hensyn til de lagrede bildedata, svarende til trinn 48. I den foretrukne
- 30 utførelse består slike adresser av x, y-koordinatene for nedre venstre piksel i rekkeviddeblokken med hensyn til matrisen av samtlige lagrede bildedata. Hvis f.eks. bildedataene lagres som en matrise av piksler med en bredde på 256 piksler og en høyde på 256 piksler og omdannes til rekkeviddeblokker, tilordnes rekkeviddeblokken i nedre venstre del av matrisen en adresse på
- 35 0,0. Tilsvarende tilordnes rekkeviddeblokken i øvre venstre del av matrisen en adresse på 0,128, den nedre høyre rekkeviddeblokken tilordnes av en

adresse 128,0 og den øvre høyre rekkeviddeblokk tilordnes en adresse på 128,128.

Ytterligere rekkeviddeblokker 38 frembringes også i den foretrukne utførelse ved å behandle rekkeviddeblokkene med lokale, affine transformasjoner såsom en rotasjon over 90° , 180° og 270° og inversjon. I tillegg til krympede rekkeviddeblokker 42 skaffes det en utbryterblokk 95 (fig. 9) som har samme størrelse som de krympede rekkeviddeblokker, men som ikke har noe innhold (dvs. det tegnes ikke noe bilde på den).

Metoden for å frembringe krympede rekkeviddeblokker skal nå beskrives mer detaljert med henvisning til fig. 7 og 8. Fig. 7 viser en første utførelse av en metode for å frembringe en krympet rekkeviddeblokk. Som det kan ses på fig. 7, innbefatter en rekkeviddeblokk 36 en rekke svart-hvittpiksler 32. I metoden vist på fig. 7, kjent som en direkte krympemetode, utgjør den krympede rekkeviddeblokken 42 utgangen i metoden og rekkeviddeblokken 36 inngangen. Den krympede rekkeviddeblokk 42 fås av rekkeviddeblokken 36 ved den følgende matematiske operasjon:

$$\text{output}(i, j) = \text{input}(2i, 2j)$$

hvor i og j henholdsvis representerer x - og y -posisjonene for piksler, vist på fig. 7.

Tilsvarende illustrerer fig. 8 en krymping ved midlings metoden, hvormed en krympet rekkeviddeblokk 42' fås fra utgangen i metoden når rekkeviddeblokken 36' leveres som inngang. Mer bestemt kan krymping ved midlingsmetoden uttrykkes matematisk som:

$$\begin{aligned} \text{output}(i, j) = & f(\text{input}(2i, 2j) + \text{input}(2i + 1, 2j) + \\ & \text{input}(2i, 2j + 1) + \text{input}(2i + 1, 2j + 1)) \end{aligned}$$

hvor: $f(0) = 0$; $f(1) = 0$; og $f(2) = f(3) = f(4) = 1$.

Entydige identifikatorer blir deretter tilordnet de avbildete rekkeviddeblokker. Hver av identifikatorene spesifiserer både adressen for undermengden av billedata hvorfra den tilsvarende rekkeviddeblokk blir frembragt og prosedy-

ren som ble benyttet på den avbildete rekkeviddeblokk. I den foretrukne utførelse blir hver av de krympede rekkeviddeblokker 42 og utbryterblokken 95 tilordnet en identifikator. Utbryterblokken har en entydig identifikator. Krympede rekkeviddeblokkidentifikatorer innbefatter en adresse og en
5 indikasjon på affine transformasjoner som er blitt benyttet på rekkeviddeblokken hvorfra den krympede rekkeviddeblokk ble utledet.

I den foretrukne utførelse består hver rekkeviddeblokk 36 av en kvadrant av 256 x 256-matrisen som utgjør de lagrede bildedata. Adressen til identifika-
10 toren, som angitt ovenfor, er x, y-koordinaten i den lagrede bildematrise for pikselet i nedre venstre hjørne av rekkeviddeblokken hvorfra den tilsvarende krympede rekkeviddeblokk ble utledet. Identifikatoren innbefatter i den foretrukne utførelse også en første kode som angir en rotasjon på 0° , 90° , 180° eller 270° og en annen kode som angir hvorvidt en inversjon er blitt
15 utført for å danne en rekkeviddeblokk forut for krympingen. Det kan ses at prosedyrespesifikasjon innbefatter tilfellet hvor det ikke benyttes noen rotasjon eller inversjon forut for krympingen.

Dernest blir tilsvarende trinn 50 på fig. 5 de avbildete rekkeviddeblokker sammenlignet med domeneblokkene. For hver domeneblokk blir den av de
20 avbildete rekkeviddeblokker valgt som mest nøyaktig i henhold til forhåndsbestemte kriterier tilsvarende denne domeneblokk.

I den foretrukne utførelse blir hver domeneblokk 34 forbundet med enten en enkelt krympet rekkeviddeblokk eller utbryterblokken. Dette oppnås ved
25 hjelp av en blokkomparator 66 (fig. 9) som fra de krympede rekkeviddeblokker 42 og utbryterblokken 95 velger den som er "nærmest" domene blokken, dvs. mest lik domeneblokken.

Selv om adressene for rekkeviddeblokkene i den foretrukke utførelse er x, y-pikselkoordinater, vil av forklaringsmessige årsaker rekkeviddeblokkadressene uttrykkes som kvadranttall 1-4, som vist på fig. 9. Tilsvarende vil
30 rotasjonen på fig. 9 bli uttrykt i grader fremfor i form av en kode. Med eksempelet vist på fig. 9, er den krympede rekkeviddeblokk nr. 2 den krympede rekkeviddeblokk som er nærmest inngangsdomeneblokken 34. På
35 tilsvarende måte vil hver domeneblokk av bildet 30 behandles av blokkompa-

ratoren 66 for å velg den nærmeste krympede rekkeviddeblokk 47 (eller utbryterblokk 95).

5 Blokkomparatoren 66 på fig. 9 skal nå beskrives mer detaljert. Formålet med blokkomparatoren 66 er for hver domene blokk å bestemme hvile av de krympede rekkeviddeblokker som utseendemessig er mest lik den spesifiserte domeneblokk. Dvs. hvilken av de krympede rekkeviddeblokker som er "nærmest" domeneblokken. Dette alternative utsagn innebærer en analogi som er nyttig for å forstå begrepet sammenligninger mellom blokker av 10 billedata. Dette begrepet betegnes som "avstand". For å evaluere "avstand" mellom blokker, skal begrepet avstand mellom punkter i rommet betraktes. Hvis to punkter A og B foreligger i et todimensjonalt koordinat system, kan deres posisjoner i koordinatsystemet henholdsvis representeres av x_A, y_A og x_B, y_B . Avstanden mellom punktene er således $\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$. Denne 15 prosedyren gir avstanden i rett linje mellom punktene A og B og betegnes som den euklidske avstand.

En annen måte å betrakte avstanden mellom punktene A og B nevnt ovenfor, er å ta i betraktning at slike punkter representerer steder i en stor by bestående av en matrise av ortogonale gater. Avstanden mellom punktene A og B 20 kunne da bestemmes som "drosjeavstanden" i stedet for "luftlinjen" og gjengis som $(|x_A - y_B|) + (|y_A - y_B|)$. En slik avstand er blitt betegnet "manhattan"-avstanden. 25

Ut fra begrepet avstanden mellom to punkter skal begrepet avstand mellom et enkelt punkt og en mengde av en rekke andre punkter betraktes. Med bruk av et hensiktsmessig avstandsbegrep, så som den ovenfor omtalte euklidske avstand eller manhattan-avstanden, kan avstanden mellom et punkt og en 30 mengde ses som summen av avstander fra punktet til hvert enkelt punkt i mengden.

Med benyttelse av begrepet avstand fra et punkt til en mengde, kan avstanden mellom to mengder av punkter betraktes. En meget generell beskrivelse av 35 avstanden mellom to mengder er en "sum av en sum". Det vil si at et første punkt i den første mengde velges. Avstanden mellom dette punkt og hvert punkt i den annen mengde beregnes. Derneft velges et annet punkt i den

første mengde. Avstanden mellom dette annet punkt og hvert punkt i den annen mengde blir deretter beregnet og addert til den første mengde. På tilsvarende måte blir avstandene mellom hvert påfølgende punkt i den første mengde og alle punktene i den annen mengde beregnet og summert for å

5 skaffe avstanden mellom de to mengder av punkter.

Begrepet avstander mellom to mengder benyttes til å beregne en avstand mellom to grupper av bildedata med det formål å velge fra en gruppe av krympede rekkeviddeblokker den krympede rekkeviddeblokk som er mest lik

10 en domeneblokk. Spesielt gjør den foretrukkede utførelse bruk av et matematisk begrep kjent som hausdorffavstanden. Dette begrepet er vel kjent i teknikken og er f.eks. omtalt i boken "Fractals Everywhere" av Michael F. Barnsley.

15 Hausdorffavstanden H mellom to blokker av bildedata f og g kan uttrykkes som H (blokk f , blokk g). I den foretrukkede utførelse beregner blokkkomparatoren 66 størrelsen H mellom en domeneblokk og hver av de krympede rekkeviddeblokker. Den krympede rekkeviddeblokk som gir den laveste verdi for H , velges som den krympede rekkeviddeblokk som er

20 "nærmest" den spesifiserte domeneblokk.

Dette begrepet kan lettere skjønnes med henvisning til fig. 10 hvori spaltene (a) og (b) henholdsvis viser to blokker av svart-hvitt bildedata, nemlig blokk f og blokk g . Blokk f består av et kryss av svarte piksler pluss et ytterligere svart piksel i øvre høyre hjørne. Blokken består av et kvadrat dannet av en

25 blokk av seksten svarte piksler. Rett under bildeblokkene f og g på fig. 10 has binære gjengivelser av bildedataene i henhold til blokkene f og g . Spesifikt består den binære gjengivelse av blokkene f og g av en matrise av nuller (0) og enere (1) hvor nullene representerer et hvitt piksel og enerne et

30 svart piksel.

Rett under hver binærblokk f og g has en avstandstabell for den respektive binære blokk. Avstandstabellen består av en matrise av tall som svarer til et av pikslene i blokkene f og g . Tallet i hver posisjon i avstandstabellen for

35 bilde blokkene f og g representerer antall piksler mellom pikselet svarende til denne posisjon og den nærmeste posisjon svarende til en svart piksel (eller i den binære blokk a 1). F.eks. er innførselen øverst til venstre i avstandstabel-

len f på fig. 10a 4. Dette gjengir det faktum at ved bevegelse bare i horison-
tal- og vertikal retningene, ikke i diagonal retningen (begrepet manhattan-
avstanden) er det fire piksler mellom pikselet øverst til venstre i bildeblok-
kene og det nærmeste svarte piksel. Dette kan ses ved å begynne ved pikselet
5 øverst til venstre, telle tre piksler til høyre og ned et piksel.

Innførselen rett til høyre for den venstre innførsel i avstandstabellen for
blokk f er tallet 3. Dette angir det faktum at ved start fra pikselet rett til
høyre for pikselet øverst til venstre i blokk f, er det tre piksler mellom det
10 angitte piksel og det nærmeste svarte piksel. Dette kan ses ved å telle to
piksler til høyre og ned et piksel. På tilsvarende vis blir hver innførsel i
avstandstabellen for blokk f bestemt. En avstandstabell blir også beregnet på
samme måte for bildeblokken g.

15 De nederste blokker på fig. 10 representerer snittet mellom avstandstabellen f
og binærblokken g og snittet mellom avstandstabellen g og binærblokken f.
En verdi D beregnes lik summen av tall i de to snittfigurer. Slik det kan
finnes av fig. 10, er summen av tallene i snittet mellom avstands tabell f og
binærblokk g 25 og summen av tallene i snittet mellom avstandstabellen g og
20 binærblokk f er 8, hvilket gir en verdi D som er lik 33. En modifisert
hausdorff-avstand H for bildeblokker f og g kan uttrykkes som følger:

$$H(\text{blokk f, blokk g}) = D(\text{blokk f, blokk g}) + D(1\text{-blokk f, } 1\text{-blokk g}).$$

25 Begrepet "1-blokk f" er den inverse av bildeblokken f og således en
bildeblokk hvori hvert piksel er det motsatte av et tilsvarende piksel i blokk
f.

Blokkkomparatoren 66 (fig. 9) beregner størrelsen H mellom en valgt
30 domeneblokk og hver krympet rekkeviddeblokk og velger som den utvalgte
krympede rekkeviddeblokk den krympede rekkeviddeblokk som gir den
laveste verdi av H.

Alternative utførelser er naturligvis mulige. F.eks. kunne en modifisert
35 hausdorff-avstand mellom blokkene beregnes med bruk av begrepet euklidisk
avstand fremfor manhattan-avstanden. Tilsvarende kunne blokkkomparatoren

66 benytte andre avstandsmålebegreper enn hausdorff-avstanden for å velge en krympet rekkeviddeblokk som er mest mulig lik hver domene blokk.

5 Straks valget av avbildete rekkeviddeblokker er fullført, blir bildeinforma-
sjonen deretter gjengitt som en mengde av entydige identifikatorer svarende
til de valgte avbildete rekkeviddeblokker, svarende til trinn 52 på fig. 5.
Listen av identifikatorer i de valgte krympede rekkeviddeblokker 42
(herunder både adresser og lokale affine transformasjonsprosedyrer, så som
rotasjon, refleksjon eller inversjon, som skal utføres på dataene spesifisert av
10 adressen) utgjør den "fraktale transformasjon" 98 av bildet 30, som vist på
fig. 11. Hver domeneblokk 34 av bildet 30 er representert ved en adresse 99
for en krympet rekkevidde blokk. F.eks. kan domeneblokken i øvre venstre
hjørne av bildet 30 være nærmest den krympede rekkeviddeblokk nr. 3.
Domeneblokken rett til høyre for øvre venstre domeneblokk kan være
15 nærmest den krympede rekkeviddeblokk nr. 5 dreid 90° . Den neste domene-
blokk straks til høyre kan være den nærmeste til utbryterblokken 95. I et slikt
eksempel ville de første tre adresser i den fraktale transformasjon 98 være
3,5(90), E som vist på fig. 11. Mengden av identifikatorer kan således lagres,
sendes eller analyseres.

20 Fig. 12 viser i detalj en foretrukket utførelse av en fremgangsmåte til
automatisk å danne en fraktaltransformasjon som representerer svart-hvitt
billedata i henhold til den foreliggende oppfinnelse. Ved trinn 100 lagres
digitale billedata i bufferen A i en dataprosessor. Fortrinnsvis består
25 billedataene av binær informasjon som representerer et svart-hvitt bilde. En
rekkeviddeblokkstørrelse bør for håndvelges slik at når inngangsbildet
lagres i buffer A, blir rekkeviddeblokkene automatisk tilordnet de forskjel-
lige digitale billedata i buffer A.

30 I trinn 102 blir rekkeviddeblokkpekere for buffer A og krympede rekkevid-
deblokkpekere for buffer B initialisert. Størrelsen av inngangsbilledatamatri-
sen og størrelsene av både rekkeviddeblokken og de krympede rekkevidde-
blokker er forhåndsbestemte verdier. Derfor kan disse pekere som består av
x,y-koordinater i bufrene A og B bestemmes.

35 I trinn 104 benyttes en krympeprosess på rekkeviddeblokken i buffer A
spesifisert av rekkeviddeblokkpekeren. I visse anvendelser kan lokale affine

transformasjoner, såsom rotasjon, refleksjon eller inversjon, benyttes på rekkevidde blokken forut for krymping for å få et større antall krympe de rekkeviddeblokker for et gitt inngangsbilde. Den krympede rekkeviddeblokken blir deretter lest inn i posisjoner i buffer B identifisert av de krympede
5 rekkeviddeblokkpekere.

Rekkeviddeblokkpekerne for buffer A og de krympede rekkeviddeblokkpekerne for buffer B blir deretter øket i trinn 106. Det foretas en bestemmelse i trinn 108 for å se hvorvidt alle rekkeviddeblokkene i buffer A er blitt
10 krympet. Hvis ikke, blir trinnene 104 og 106 gjentatt utført inntil alle rekkeviddeblokkene i buffer A er blitt krympet.

Deretter blir en domeneblokkpeker i buffer A initialisert (trinn 112) og den
15 krympede rekkeviddeblokkpeker i buffer B initialiseres (trinn 114). I trinn 116 blir hausdorff-avstanden mellom domeneblokken spesifisert av domeneblokkpekeren og den foreliggende krympede rekkeviddeblokk spesifisert av den krympede rekkeviddeblokkpeker, bestemt.

20 Hvis hausdorff-avstanden som nettopp ble beregnet, er mindre enn den for øyeblikket lagrede avstandsverdi (som initialiseres til den størst mulige hausdorff-avstand), så vil i trinn 118 den forekommende lagrede avstandsverdi erstattes av den avstandsverdi som sist ble beregnet. Den nåværende krympede rekkeviddeblokkpekerverdi blir også lagret. Hvis den aller sist
25 beregnede hausdorffavstand ikke er mindre enn den for øyeblikket lagrede avstandsverdi, blir den aller sist beregnede avstandsverdi forkastet.

I trinn 120 blir den krympede rekkeviddeblokkpeker øket, og det gjøres en bestemmelse av hvorvidt alle de krympede rekkeviddeblokker er blitt sjekket.
30 Hvis ikke, gjentas trinnene 116 og 118 inntil alle krympede rekkeviddeblokker er blitt sjekket. Om så er, blir den for øyeblikket lagrede, krympede rekkeviddeblokkpeker, bestående av x, y-koordinater og den forbundne prosedyreinformasjon (refleksjon og inversjon) så vedføyd listen av identifikatorer som representerer domeneblokkene for inngangsbilddataene lagret i
35 buffer A (trinn 124).

Deretter blir domeneblokkpekeren øket i trinn 126 og det gjøres en bestemmelse av hvorvidt alle domeneblokkene er blitt behandlet. Hvis ikke, gjenntas trinnet 116-126 inntil alle domeneblokkene er blitt behandlet. Listen av identifikatorer som hver svarer til en av domeneblokkene for de opprinnelige inngangsdata, blir deretter gitt ut i trinn 130 som fraktaltransformasjonen av bildedataene som representerer svart-hvitt bildet levert som inndata til dataprosessoren.

I den foretrukkede utførelse blir identifikatorene for de krympede rekkeviddeblokker uttrykt som følger:

$$XYrri$$

hvor:

X er en åtte-biters verdi som representerer x-koordinaten for det nedre venstre hjørne av rekkeviddeblokken i det opprinnelige bildematrikse,

Y er en åtte-biters verdi som representerer Y-koordinaten i nedre venstre hjørne av rekke viddeblokken i den opprinnelige bildematrikse,

rr er en to-biters verdi som representerer rotasjon som følger:

00 = ingen rotasjon

01 = 90° rotasjon

10 = 180° rotasjon

11 = 270° rotasjon,

i er en en-bits verdi som representerer inversjon som følger:

0 = ingen inversjon og

1 = n=inversjon.

I henhold til et annet trekk skaffer den foreliggende oppfinnelse en fremgangsmåte til å generere et bilde med en dataprosessor fra et komplett datasett bestående av en mengde av identifikatorer. Dataprosessoren innbefatter et minne med en rekke buffere som hver har en rekke adresser bare områder spesifisert av en bufferadresse. Hver av identifikatorene svarer til et adresserbart område i bufferne og innbefatter en bufferadresse og en prosedyrespesifikasjon. Som vist på fig. 13, innbefatter fremgangsmåten et første trinn 140 som omfatter å legge inn mengden av identifikatorer. Trinn 142 omfatter å lagre et forhåndsbestemt mønster av digitale data i en av bufferne betegnet som en kildebuffer. Trinn 144 består i for hver av identifikatorene å bestemme et mønster av data som svarer til hver identifika-

tor, ved å benytte prosedyren spesifisert i denne identifikator og den del av kildebufferen som indikeres ved bufferadressen i denne identifikator.

Deretter omfatter et trinn 146 å lagre mønstre av data i det adresserbare område av en annen av bufferne, betegnet som en destinasjonsbuffer, angitt i den tilsvarende identifikator. Et trinn 148 omfatter å gjenta lagringstrinnene for bestemmelsen og destinasjonsbufferen med destinasjonsbufferen betraktet som kildebuffer inntil de forhåndsbestemte kriterier oppfylles. Endelig omfatter et trinn 149 å skaffe innholdet i destinasjonsbufferen som data som representerer et bilde når de forhåndsbestemte kriterier er oppfylt.

Fig. 14 viser en dataprosessor 150 for å eksekvere fremgangsmåten på fig. 13. Dataprosessoren 150 innbefatter en inngangsinnretning såsom en datamottager eller en plate stasjon. Inngangsinnretningen 152 er forbundet til en CPU 154 som innbefatter en kontrollogikk 156, en teller 158 og en mønstergenerator 160. En minneinnretning 162 er forbundet til CPU 154 og innbefatter et par av buffere 164 og 166. En utgangsinnetning 168 er også forbundet til CPU 154 og kan omfatte en CRT-visningsinnretning.

Fig. 15 viser et system for å utføre en foretrukket fremgangsmåte i henhold til oppfinnelsen for å gjenvinne bildet fra den fraktale transformasjon 98. Først lagres et forhåndsbestemt mønster av digitale data i en av bufferne betegnet som en kildebuffer. Som vist på fig. 15, blir digitale data som gjengir et initialt bilde 202, og den fraktale transformasjonen 198 levert til et dekompresjons system 204 som fortrinnsvis omfatter dataprosessoren 150. Det initiale bilde 202 lastes i en buffer A, betegnet som kildebufferen.

Dekompresjonssystemet 204 bestemmer deretter for hver av identifikatorene tilsvarende mønster av data. Det tilsvarende mønster blir bestemt ved å benytte prosedyren spesifisert av denne identifikator på delen av kildebufferen som angis med identifikatorens bufferadresse. Dekompresjonssystemet 204 lagrer deretter det resulterende mønster i et tilsvarende adresserbart område av en annen av bufferne, buffer B som betegnes som en destinasjonsbuffer.

Denne prosedyren er vist på fig. 16. På denne figur virker en buffer A som en kildebuffer og fylles med data som representerer det initiale bilde 202. En buffer B virker som en destinasjonsbuffer og deles i dekodedomeneblokker

206. Hver av dekodedomeneblokkene 206 svarer til en identifikator for den innlagte fraktale transformasjon 198. Ved bestemmelsen av datamønstrene som skal lagres i hver dekodedomeneblokk 206, blir den tilsvarende identifikator i domeneblokken benyttet. Som beskrevet ovenfor, består hver

5 identifikator i den fraktale transformasjon 198 av en adresse og en prosedyrespesifikasjon. Adressen representerer en rekkeviddeblokk utledet fra en spesifisert del (f.eks. en nedre venstre kvadrant) av et sett av data som representerer et inngangs bilde. I dekompresjonsprosessen blir adressen til identifikatorene svarende til en dekodedomeneblokk 206 benyttet til å

10 spesifisere den samme del, f.eks. den nedre venstre kvadrant, av datasettet lagret i kildebufferen, buffer A på fig. 16, som representerer det initiale bilde 202. Denne del av dataene lagret i buffer A blir deretter behandlet med prosedyren spesifisert av identifikatoren svarende til domene blokken og de behandlede data ytterligere behandlet ved krymping. De behandlede og

15 krympede data blir deretter lagret i en tilsvarende dekodedomeneblokk i buffer B.

Hvis en identifikator svarende til en spesifisert dekodedomeneblokk 206 i buffer B f.eks. består av "1(90)", representerer dette en adresse og en

20 prosedyre. Adressen angir at rekkeviddeblokken # 1 er involvert, og denne består av en nedre venstre kvadrant av et bilde. Prosedyren spesifiserer rotasjon med 90°. For således å utlede de digitale data som skal lagres i den spesifiserte dekodedomeneblokk, blir dataene som representerer nedre venstre kvadrant i buffer A behandlet av en prosedyre for å rotere det

25 tilsvarende bildet med 90° og dataene blir deretter behandlet med en krympealgoritme for å danne en krympet rekkeviddeblokk på måten omtalt ovenfor med hensyn til prosessen for å danne den fraktale transformasjon. Det resulterende data som representerer den krympede rekkeviddeblokk, blir deretter lagret i den spesifiserte dekodedomeneblokk i buffer B.

30 På tilsvarende måte blir hver av dekodedomeneblokkene 206 fylt med data spesifisert av den tilsvarende identifikator for den fraktale transformasjon 98. Hvis adressen i den tilsvarende identifikator spesifiserer utbryterblokken, dvs. en blokk som inneholder bildedata som representerer et tomt bilde, blir

35 en utbryterblokk lagret i en tilsvarende dekode domeneblokk.

I den foretrukkede utførelse av oppfinnelsen blir de foregående to trinn, nemlig (1) fylling av bufferen A med bilde data og (2) fylling av alle dekodedomeneblokkene i buffer B med digitale data som representerer de krympede rekkeviddeblokker i buffer A, spesifisert av identifikatorene i den fraktale transformasjon, gjentatt inntil forhåndsbestemte kriterier er oppfylt, som vist på fig. 18. Under denne gjentakelse blir bilde dataene i buffer B på fig. 16 i stedet for å fylle buffer A med data som representerer det initiale bilde 202, lest tilbake til buffer A. Dette kan ses ved å bemerke at innholdet i buffer B på fig. 16 er lagret i buffer A på fig. 17.

Fremgangsmåten slutter med å skaffe innholdet i destinasjonsbufferen som data som gjengir et bilde når de forhåndsbestemte kriterier er oppfylt.

Således gjentas trinnene i fig. 16 og 17 et bestemt antall ganger inntil "konvergens" er oppnådd. Antallet iterasjoner som er overvåket av telleren 158 (fig. 14) avhenger av dimensjonene (målt i piksler) av domeneblokkene og rekkeviddeblokkene og kan bestå av et fast antall iterasjoner. Alternativt kan iterasjonene være bestemt ved å måle forskjellen mellom bilde dataene i bufferne A og B og avslutte iterasjonsprosessen når forskjellen er mindre enn den for håndsbestemte verdi.

Fig. 19 viser en detaljert beskrivelse av en foretrukket fremgangsmåte for automatisk å generere et bilde med en dataprosessor i henhold til den foreliggende oppfinnelse fra et sett av identifikatorer bestående av den fraktale transformasjon av et originalbilde. Den fraktale transformasjon kunne genereres ved fremgangsmåten på fig. 12 eller enhver annen fremgangsmåte som kan generere et sett av identifikatorer så som frembragt ved fremgangsmåten på fig. 12.

I trinn 250 blir en buffer A som består av en bildebuffer i dataprosessoren, initialisert med initiale bilde data som representerer et fast svart bilde (dvs. alle enere). Ethvert initialt bilde kunne benyttes. Imidlertid er et helsvart bilde en hensiktsmessig og lett lagret mengde av bilde data.

I trinn 252 blir en hovedsløyfeteller (hvis funksjoner kan skaffes med telleren 158) initialisert til null. Hoved sløyfetelleren angir antall ganger bildegenereringstrinnene på fig. 16 og 17 er blitt gjentatt. Dette tallet kan

sammenlignes med en forhåndsbestemt verdi valgt for den bestemte anvendelse for å avslutte iterasjonsprosessen.

5 I trinn 254 blir en domeneblokkpeker initialisert for en annen buffer, buffer B, som tidligere er blitt delt i domeneblokker. I trinn 254 blir også en fraktal transformasjonspeker initialisert for å peke på den første identifikator i den fraktale transformasjon som representerer de innlagte komprimerte bildedata.

10 I trinn 256 blir identifikatoren spesifisert av den fraktale transformasjonspeker benyttet på bildedata i buffer A. Adressen i denne identifikator spesifiserer en rekkevidde blokk i buffer A, og prosessen til denne identifikator blir deretter benyttet på den spesifiserte rekkeviddeblokk. Som tidligere bemerket, kunne identifikatoren også spesifisere utbryterblokken. Derne-
15 st blir en krympet rekkeviddeblokk konstruert fra den behandlede, spesifiserte rekkeviddeblokk fra buffer A og den krympede rekkeviddeblokk (eller utbryterblokken) leses inn i en domeneblokk i buffer B som blir pekt på av den foreliggende verdi av domene blokkpekeren. Teknikken for å skaffe en krympet rekkevidde blokk kan benytte hvilken som helst av krympemetodene som er beskrevet ovenfor, eller kan benytte en helt annen metode.

20 I trinn 258 blir domeneblokken og de fraktale transformasjonspekere øket. Det gjøres en bestemmelse i trinn 260 av hvorvidt den siste identifikator i den innlagte fraktale transformasjon er blitt behandlet. Hvis ikke, går fremgangsmåten i sløyfe tilbake til trinn 256.

25 Hvis den siste fraktale transformasjonsidentifikator er blitt behandlet, skrider fremgangsmåten frem til trinn 262 hvor hovedsløyfetelleren økes. Det gjøres en bestemmelse i blokk 264 av hvorvidt hovedsløyfetelleren har nådd en maksimumsverdi. Om ikke, overføres innholdet i buffer B til buffer A som
30 vist ved trinn 266 og metoden går i sløyfe tilbake til trinn 254. Etter et forhåndsbestemt antall iterasjoner av trinnene 254-266, leveres innholdet i buffer B som utgangsdata som representerer det opprinnelige ukomprimerte bilde.

35 Fremgangsmåten vist på fig. 19, skaffer en iterativ prosess hvori antallet iterasjoner er spesifisert med en forhåndsbestemt verdi. Alternativt kunne iterasjonene ved fremgangsmåten vist på fig. 19 fortsette inntil innholdet i

buffer B bare skiller seg fra innholdet i buffer A med mindre enn en forhåndsbestemt verdi. Denne fremgangsmåten, kjent som konvergensmetoden, kan ytterligere beskrives som følger. Etter trinnet hvor innholdet i buffer B er blitt beregnet, utføres de følgende trinn:

5

i) initialisering av A og B, henholdsvis bestående av en peker til en domeneblokk i buffer A og en peker til en domeneblokk i buffer B som har den samme relative posisjon i buffer B som domeneblokken i buffer A som pekes på av A.

10

ii) initialisere et heltallregister til null.

iii) benytte en blokkomparator, så som komparatoren 66, for å beregne en avstand mellom domeneblokken pekt på av A og domeneblokken pekt på av B.

15

iv) øke registerverdien med denne avstand.

v) øke pekerne a og b.

vi) gjenta trinnene (iii) og (iv) inntil alle domene blokkene er blitt behandlet.

20

vii) lagre registerverdien som den endelige registerverdi.

viii) gjenta trinnene 1-7 hver gang innholdet i buffer B blir beregnet på ny og sammenligne de påfølgende endelige registerverdier.

(ix) avslutte iterasjonene vist på fig. 19 når de påfølgende endelige registerverdier har en forskjell som er mindre enn en forhåndsbestemt terskelverdi.

25

Slik det vil være innlysende for vanlige fagfolk, kunne ekvivalente implementeringer av den foreliggende oppfinnelse gis. F.eks. kunne originalbildet genereres fra den komprimerte fraktale transformasjon bare med bruk av en enkelt bildebuffer som vist på fig. 20. Fig. 20 identifiserer trinnet i en

30

alternativ utførelse av oppfinnelsen for en fremgangsmåte til å generere et bilde med en dataprosessor fra et sett av identifikatorer. Dataprosessoren innbefatter en buffer med en rekke adresserbare områder spesifisert av en bufferadresse. Hver av identifikatorene svarer til et adresserbart område i bufferen og innbefatter både en bufferadresse og en prosedyre som skal anvendes på dataene i områdeidentifikatoren ved den tilsvarende adresse.

35

Først blir et forhåndsbestemt mønster av digitale data for et vilkårlig bilde lagret i bildebufferen. Dette er angitt i trinn 300 hvor billedata som represen-

terer et vilkårlig inngangsbilde lagres i en bildebuffer. Sett av identifikatorer svarer til en fraktal transformasjon og leveres som inngang til en dataprosessor som vist i blokk 304. For hvert adresserbart område av bildebufferen bestemmes et mønster av data svarende til denne identifikator ved å benytte prosedyren i denne identifikator på delen av bildebufferen angitt av bufferadressen i denne identifikator. Datamønsteret blir deretter lagt tilbake i samme område av bildebufferen, angitt av bufferadressen i den tilsvarende identifikator.

- 5
- 10 Som vist på fig. 20, utføres disse trinnene ved å innstille en peker til en første domeneblokk i bildebufferen (ved trinn 306). Deretter blir adressen til en rekkeviddeblokk svarende til bildeblokken spesifisert av pekeren lest i trinn 308. En krympet rekkeviddeblokk blir deretter konstruert fra bildedataene i bildebufferen som angitt i blokk 310. Deretter blir dataene som forekommer i
- 15 domeneblokken i identifikatoren, erstattet av data fra den krympede rekkeviddeblokk som angitt i trinn 312.

- Trinnene for å identifisere og lagre blir deretter gjentatt inntil et forhåndsbestemt kriterium er oppfylt, og innholdet i bildebufferen blir levert i form av data som representerer et bilde. Som vist i trinn 314, blir domeneblokkpekeren øket ved en bestemmelse gjort i trinn 316 om hvorvidt alle domeneblokkene er blitt behandlet. Hvis ikke, blir trinnene 308,310 og 312 gjentatt inntil alle domeneblokkene er behandlet. Det gjøres så en bestemmelse i trinn 318 om hvorvidt innholdet i bildebufferen er stabilt, dvs. skiller seg med mindre enn en forhåndsbestemt verdi fra det tidligere innhold. Denne bestemmelsen
- 20
- 25 gjøres ved å utføre trinnene (i)-(vii) i konvergensmetoden beskrevet ovenfor. Når den forhåndsbestemte terskelverdi for den endelige registerverdi er nådd, blir innholdet i bildebufferen deretter levert som utdata som representerer det opprinnelige komprimerte bilde som angitt i trinn 320.

- 30
- Den foreliggende oppfinnelse er egnet til å behandle gråskalabilder såvel som svart-hvitt bilder. Et gråskalabilde er et bilde dannet av piksler som kan ha hvilken som helst av rekke graderte lysintensiteter, som går fra fullstendig svart til fullstendig hvitt. Fig. 21-24 viser en foretrukket prosess hvormed en
- 35 fraktaltransformasjon av en innlagt digitalisert gråskalabilde 360 finnes. Bildet 360 deles i domeneblokker 362 og rekkeviddeblokker 364 som henholdsvis vist på fig. 21 og 22. Ytterligere rekkeviddeblokker kan også

dannes ved prosedyrer såsom ved rotasjon og refleksjon, som tidligere beskrevet med hensyn til svart-hvitt rekkeviddeblokker.

En krympeprosess blir deretter benyttet på rekkevidde blokkene for å frembringe krympede rekkeviddeblokker 366 som vist på fig. 23. Krympeprosessen er slik at de krympede rekkeviddeblokker inneholder samme antall piksler som domeneblokkene, og kan benyttes til krympemetoder som beskrevet ovenfor med hensyn til svart-hvitt bilder. Til forskjell fra prosedyren for å behandle svart-hvitt bilder er det imidlertid ikke innbefattet noen utbryterblokk blant de krympede rekkeviddeblokker. Hvert piksel av domeneblokkene og de krympede rekkeviddeblokkene har et til seg forbundet tall som representerer dets gråskalaverdi, dvs. lysstyrken til pikselet. F.eks. har hvert piksel i bildet 360 (fig. 21 og 22) en gråskalaverdi på 0-255.

Som vist på fig. 24, blir hver domeneblokk 362 sammenlignet med krympede rekkeviddeblokker 366 forbundet med en enkelt krympet rekkeviddeblokk 369 som er den som mest nøyaktig svarer til domeneblokken. Denne forbindelsen vil bli beskrevet mer detaljert nedenfor med hensyn til fig. 25 og utføres av en blokkomparator 368 som sammenligner gråskalaverdiene for pikslene i hver blokk med gråskalaverdier for pikslene i hver krympet rekkeviddeblokk 366. Som hjelp til å finne en krympet rekkeviddeblokk 366 som mest nøyaktig svarer til en domeneblokk, kan vertikale skaleringsparameter P og Q benyttes til prosessering av gråskalaverdier V_{GS} for domeneblokkpikslene for å finne en ny pikselgråskalaverdi $V'_{GS} = PV_{GS} + Q$.

Fig. 25 gir ytterligere forklaring på behandling av gråskalabilder. Et gråskalabilde 400 dannet av piksler 401 med fire mulige nivåer av grått, kan være representert på binær form som vist ved 402. Således har hvert piksel i den binære representasjon 402 sitt intensitetsnivå representert av en verdi på 0-3.

Krympede rekkeviddeblokker 366 kan konstrueres fra den binære representasjon 402. Først kan fire binære representasjoner av rekkeviddeblokker 364 konstrueres som nedre venstre (1), nedre høyre (2), øvre venstre (3), og øvre høyre (4) kvadrant av den binære bilderepresentasjon 402. Disse binære rekkeviddeblokkrepresentasjonene blir deretter behandlet med krympemetoder som beskrevet ovenfor, for å danne krympede rekkeviddeblokker 404.

Den binære representasjon 402 blir også delt i binære domeneblokker 406. De binære domeneblokker blir nummerert 1-16 fra venstre mot høye og fra bunn til topp i den binære representasjon 402. Hver binær domeneblokk 406 er

5 en 2 x 2 pikselmatrise av ikke-overlappende undermengder av den binære bilderepresentasjon 402. Domeneblokkstørrelsen velges som den samme som den krympede rekkeviddeblokkstørrelse.

10 Med bruk av en verdi P lik 0,5 og verdier av Q som angitt på fig. 25 for å beregne nye pikselgråskalaverdier $V'_{GS} = PV_{GS} + Q$, kan hausdorff-avstandene mellom hver binær domeneblokk 406 og hver binær krympet rekkeviddeblokk 404 beregnes og for hver binær domeneblokk 406 velges den krympede binære rekkeviddeblokk 404 som ligner mest. Resultatene er vist i tabell 408

15 på fig. 25. Tabell 408 viser at domeneblokkene 1-6, 8-10 og 14 er mest lik den krympede rekkeviddeblokk 1, domene blokkene 7,11 og 16 er mest lik den krympede rekkeviddeblokk 2 og domeneblokkene 12, 13 og 15 er mest lik den krympede rekkeviddeblokk 3. Den fraktale transformasjon 410 av bildet 400 består da av den sekvensielle liste av krympede rekkeviddeblokk-

20 tall sammen med de vertikale skaleringsparametre P og Q, som vist i tabell 408 og på fig. 26.

Den tilsvarende fraktale transformasjon 410 for det innlagte gråskalabilde 60 er således en liste av identifikatorer bestående av "adressene" til de krympede rekkeviddeblokker og forbundne vertikale skaleringsparametre, idet en

25 adresse og et sett av vertikale skaleringsparametre er assosiert med hver domeneblokk som vist på fig. 26. Selv om ingen prosedyrer så som rotasjon og inversjon, er blitt omtalt i tilknytning til fig. 25 for å unngå en unødige komplisert forklaring, benytter den foretrukne utførelse slike prosedyrer, og

30 koder som spesifiserer prosedyrene er innbefattet i hver identifikator i den fraktale transformasjon.

Fig. 27 viser detaljert en foretrukket fremgangsmåte til automatisk å frembringe en fraktaltransformasjon som representerer gråskalabildedata i

35 samsvar med den foreliggende oppfinnelse. I trinn 500 blir digitale bildedata som representerer et gråskalabilde lagret i buffer A i en dataprosessor. I trinn 502 blir en rekkeviddeblokkpeker for buffer A og en krympet rekkevid-

deblokkpeker for buffer B initialisert. I trinn 504 blir forhåndsbestemte prosedyrer benyttet på rekkeviddeblokken i buffer A spesifisert av rekkeviddeblokkpekeren. Slike prosedyrer innbefatter lokale affine transformasjoner, så som rotasjon, refleksjon eller inversjon forut for krymping for å skaffe et større antall krympede rekkeviddeblokker for et gitt inngangsbilde. De behandlede rekkeviddeblokker blir deretter lagret i buffer B i posisjoner angitt av de krympede rekkeviddeblokkpekere.

Rekkeviddeblokkpekeren for buffer A og den krympede rekkeviddeblokkpeker for buffer B blir deretter passende øket i trinn 506 og rekkevidden undersøkt i trinn 508 for å se hvorvidt alle rekkeviddeblokkene er blitt aksessert. Hvis ikke, gjentas trinnene 504 og 506 inntil alle rekkeviddeblokkene er blitt aksessert. En domeneblokkpeker for buffer A blir initialisert i trinn 512, og den krympede rekkeviddeblokkpeker for buffer B initialiseres i trinn 514.

I trinn 516 blir vertikale skaleringsparametre P og Q bestemt for å minimere L^2 -avstanden mellom domeneblokkene adressert av domeneblokkpekeren og den krympede rekkeviddeblokk adressert av rekkeviddeblokkpekeren. Som vel kjent for fagfolk, benyttes L-avstandene til å måle avstandene mellom grafene av funksjoner i stedet for mellom mengder av punkter. Et digitalisert gråskalabilde kan ses som en funksjon dannet av den todimensjonale matrise av indekser som representerer pikselposisjoner i et bilde og som angir intensitetsverdien forbundet med pikslene. Det vil si at funksjonen $f(i,j)$ representerer intensiteten ved piksel i,j hvor i,j representerer posisjonen av pikselet i bilde matrisen. En L-avstand sammenligner to funksjoner ved passende summering og midler i en viss forstand avstandene mellom intensitetene ved hver piksel. Det er en L^2 -avstand hvis forskjellene kvadreres før de summeres, i hvilket tilfelle "midlingen" oppnås ved å ta kvadratroten av summen. Det er en L^1 -avstand hvis absoluttverdien av forskjellene benyttes. Hvis L-avstanden mellom to bilder er liten, vil de to bildene se like ut. L-avstandene har den fordel at de generelt kan beregnes hurtigere og lettere enn hausdorff-avstandene. Spesielt benyttes en veiet L^2 -avstand D_L i den foretrukne utførelse for å velge vertikale skaleringsparametre P og Q og uttrykkes som:

$$D_L = \sqrt{\sum_{i=1}^N |x_i - y_i|^2 w_i}$$

hvor: x_i, y_i er indekserte indeksverdier i respektive bilder, w_i er ikke-negative tall og N er antallet piksler i hvert bilde.

5

Avstanden mellom digitale gråskalabilder A og B (eller blokker tatt fra disse) kan måles med bruk av både hausdorff-avstander og bruk av L-avstander. I det første tilfelle behandles bildene som mengder av punkter i en tredimensjonal matrise, og i det annet tilfelle behandles de som funksjoner. I utførelsen vist på fig. 27, benyttes L^2 -avstander for å velge P- og Q-verdier.

10

I trinn 518 bestemmes hausdorff-avstanden mellom domene blokken spesifisert av domeneblokkpekeren og den krympede rekkeviddeblokk spesifisert av den krympede rekkevidde blokkpeker og skalert av P og Q. Hausdorff-avstanden blir beregnet på samme måte som beskrevet i tilknytning til fig. 10. En bestemmelse gjøres i trinn 520 hvis verdien beregnet i trinn 518 er mindre enn den forekommende lagrede verdi. Om dette er tilfelle, blir den forekommende lagrede verdi erstattet av verdien nettopp beregnet i trinn 518 sammen med identifikatoren til den krympede rekkeviddeblokk som ble benyttet til å generere den nettopp beregnede verdi. Hvis verdien beregnet i trinn 518 ikke er mindre enn den for øyeblikket lagrede verdi, blir den nettopp beregnede verdi forkortet.

15

20

25

I trinn 522 blir rekkeviddeblokkpekerne øket, og det gjøres en bestemmelse av hvorvidt alle rekkeviddeblokkene er blitt sjekket. Hvis ikke, går fremgangsmåten på fig. 22 i sløyfe tilbake til trinnene 516-522 inntil alle krympede rekke viddeblokker er blitt sjekket.

30

35

Identifikatoren lagret av sløyfen bestående av trinnene 516-522, blir deretter i trinn 526 føyd til en liste av identifikatorer som hver representerer en av domeneblokkene i inngangsbilddataene lagret i buffer A. Domeneblokkpekeren blir øket i et trinn 528, og det gjøres en bestemmelse av hvorvidt alle domeneblokkene er blitt behandlet. Hvis ikke, gjentas trinnene 516-528 inntil alle domeneblokkene er blitt behandlet. Listen av identifikatorene som hver svarer til en av domeneblokkene i de opprinnelige inngangsdata, blir deretter gitt ut i trinn 532 som fraktaltransformasjonen i henhold til denne fremgangsmåten for bilddata som representerer de innlagte gråskalabilder.

Fig. 28 viser hvordan et gråskalabilde gjenvinnes fra en fraktaltransformasjon 410 (som innbefatter skaleringsparametre). Et vilkårlig initialt bilde 534 og en fraktal transformasjon 478 gis til et dekompresjonssystem 536, fortrinnsvis realisert i en dataprosessor. Initialbildet 534 leses inn i buffer A.

5 En buffer B deles i domeneblokker, og det leses i hver domeneblokk inn en krympet rekkeviddeblokk som er spesifisert av den tilsvarende adresse i den fraktale transformasjon 410, skalert som spesifisert av de vertikale skaleringsparametre. Den krympede rekkeviddeblokk konstrueres fra initialbildet 534 i buffer A. Bildet 534 i buffer A blir behandlet, adressert og omdannet til

10 krympede rekke viddeblokker på nøyaktig den samme måte som det opprinnelige bilde. Hver av identifikatorene som danner den fraktale gråskalatransformasjon, svarer til en av domeneblokkene i buffer B på fig. 28 og benyttes til å velge en krympet rekkevidde fra buffer A for innlesning i den tilsvarende domeneblokk og danner således et midlertidig bilde i buffer B. Det

15 midlertidige bilde i buffer B blir nå lest tilbake til buffer A og buffer B slettes. Trinnene for å innlese i hver domeneblokk av buffer B en krympet rekkeviddeblokk av buffer A spesifisert av den tilsvarende adresse i den fraktale transformasjon for å danne et midlertidig bilde og å lese det midlertidige bildet i buffer B inn i buffer A, gjentas enten inntil et forhåndsgitt

20 antall ganger eller inntil "konvergens" er oppnådd.

Generelt avhenger antall iterasjoner av trinnene for å plassere krympede rekkeviddeblokker i buffer B og erstatte innholdet i buffer A med innholdet i buffer B, av dimensjonene (målt i piksler) til domeneblokkene og rekke

25 viddeblokkene. Antallet iterasjoner kan også bestemmes eksperimentelt ved å detektere konvergens, definert som når de påfølgende bilder i buffer B ikke forandrer seg med mer enn en forhåndsbestemt verdi. Det dekodete bilde er et som til slutt fås etter at den iterative prosess er fullført.

30 Fig. 29 viser et logisk flytdiagram som utgjør en foretrukket utførelse for å dekode en fraktal gråskalatransformasjon så som den som ble generert ved fremgangsmåten på fig. 27. Først leveres et initialbilde til buffer B som angitt ved trinn 550. Derneft blir en hovedsløyfeteller, en domene blokkpeker for buffer B og en fraktaltransformasjonspeker initialisert, som angitt i trinn

35 552. Den neste innførsel av den fraktale transformasjon som spesifisert av den fraktale transformasjonspeker, fås i trinn 554 og en tilsvarende krympet rekkeviddeblokk dannes fra buffer A. Rekke viddeblokken ble deretter

skalert med vertikale skalerings parametre P og Q og lest inn i en domeneblokk med en tilsvarende posisjon i buffer B. I trinn 556 blir domeneblokk pekeren for buffer B og den fraktale transformasjonspeker øket. En bestemmelse gjøres i trinn 558 av hvorvidt den siste domeneblokk og den fraktale transformasjon er blitt behandlet. Hvis ikke, går fremgangsmåten i sløyfe tilbake gjennom trinnene 554 og 556 inntil hver innførsel av den fraktale transformasjon er blitt behandlet.

Hovedsløyfetelleren økes i trinn 560. Hvis det foreskrevne antall iterasjoner ikke er blitt nådd (trinn 562), går metoden i sløyfe tilbake til trinn 552 og danner en domene blokk i buffer B fra hver innførsel av den fraktale transformasjon. Prosessen fortsetter inntil det foreskrevne antall iterasjoner er nådd. Samtidig skaffes innholdet i buffer B som data som representerer gråskalabildet som tidligere ble komprimert i den innlagte fraktale transformasjon.

Et numerisk eksempel på konstruksjonen av en fraktal gråskalatransformasjon som benytter vertikale skaleringsparametre, skal nå beskrives. Med bruk av prosedyren vist på fig. 27 med 8 x 8 domeneblokker og 16 x 16 rekkeviddeblokker, ble 256 x 256-gråskalaen omformet til digital data. Det opprinnelige billedata krevde et 65 536 byters minne for lagring. Etter behandlingen som benyttet fremgangsmåten på fig. 27, ble det dannet en fraktal transformasjon som krever et 2560 byters minne for lagring.

Andre størrelser på domeneblokker og rekkeviddeblokker, såvel som forskjellige antall rekkeviddeblokker kunne naturligvis benyttes. Reduksjon av blokkstørrelsen øker antallet blokker og øker påliteligheten av kompresjons/dekompresjonsprosessen. Det vil si at forskjellen mellom de opprinnelige og dekodete bilder reduseres. Eksakt kompresjon fås når det ikke har noen forskjell mellom originalbildet og det dekomprimerte bilde som betraktet ved bestemte skjermopløsning. Reduksjon av blokkstørrelsen og økning av antallet mulige rekkeviddeblokker øker imidlertid også lengden av søket når den nærmeste krympede rekkevidde blokk skal bestemmes. Dette øker i sin tur behandlingstiden for en gitt datamaskintype eller øker kostnadene hvis en kraftigere datamaskin benyttes for å eliminere den potensielt lengre behandlingstid.

Den foreliggende oppfinnelse skaffer også en fremgangsmåte til å forsterke bilder. Billedata som representerer et gråskalabilde med en første oppløsning, lagres i dataprosessoren. F.eks. leveres data fra et bilde bestående av matrise på 256 x 256 piksler som inndata ved fremgangsmåten på fig. 27. En fraktaltransformasjon fås ved å benytte denne fremgangsmåten. Den fraktale transformasjon blir deretter dekodet med en annen oppløsning som er høyere enn den første oppløsning, for å danne utgangsbilledata ved dekoding av den fraktale transformasjon med bruk av fremgangsmåten på fig. 29, innbefattet et par av større buffere for å danne en utgangsbildematrikse som er større enn 256 x 256. F.eks. gir bruk av bufferne i fremgangsmåten på fig. 29 for å skaffe en 512 x 512 utgangsmatrikse, et bilde som har en oppløsning som er den dobbelte av det opprinnelige inngangsbilde. Hvis en skriftsort benyttes til å danne et sett av inngangsbilder, kan denne metoden benyttes til å skaffe et skalerbart typesnitt. Det vil si at de fraktale transformasjoner som representerer symbolene i skriftsorten, kan dekodes til enhver ønsket oppløsning for å skaffe multiple typesnitt for skriftsortsymbolene med en hvilken som helst ønsket størrelse.

Oppfinnelsen kan også benyttes til å komprimere og dekomprimere fargebilder som vist på fig. 30. Digitale data som representerer et fargebilde legges inn, og det dannes en trio av fraktale gråskalatransformasjoner, en for hver av primærfargene rød, grønn og blå. De tre fraktale transformasjonene blir sammenkjedet for å danne en fraktal farge transformasjon. For å gjengi de opprinnelige fargebilder fra de komprimerte data i den fraktale fargetransformasjon, deles den fraktale fargetransformasjon i individuelle fraktale transformasjoner som representerer fraktale grå skalatransformasjoner av røde, grønne og blå bilder. De individuelle fraktale transformasjoner blir deretter dekodet med f.eks. bruk av fremgangsmåten på fig. 29 for å danne røde, grønne og blå intensitetsbilder. Disse intensitetsbildene kombineres på en måte vel kjent i teknikken og gis ut som et dekomprimert fargebilde.

Mens algoritmene for dekoding av den fraktale gråskala transformasjon kan implementeres i en lang rekke alminnelige datamaskiner, vil det skjønnes at den lille, påkrevde variasjon i operasjonene og et stort antall operasjoner effektivt kan utføres i en spesiell beregningsinnretning. Følgelig er et foretrukket apparat til å behandle digitale billedata og som benytter fremgangsmåten beskrevet ovenfor, vist på fig. 31.

Apparatet på fig. 31 innbefatter et vertsgrensesnitt 600, minner 602, 604, 605 og 606, en minnegrensesnittkrets 610, en digital signalprosessor 612 og et visningsutgangsgrensesnitt 613 på et trykt kretskort som er kompatibelt med datamaskiner av PC/AT-typen basert på sentralprosessen Intel 80286.

Den digitale signalprosessor 612 og minnegrensesnittet 610 er forbundet til en PC-ekspansjonsbuss 611 for å motta kommandoer og data. Den foretrukke digitale signal prosessor 612 er en TMS320C30 fremstilt av Texas Instruments, Incorporated. Detaljer vedrørende virkemåten til den foretrukke digitale signalprosessor 612 er tilgjengelig i litteraturen levert av produsenten.

Det vil skjønnes at mens den digitale signalprosessor 612 benyttet i den viste utførelse, er i stand til å skaffe en lang rekke flytetalloperasjoner, er det formål som den benyttes til i den foreliggende oppfinnelse å skaffe høy hastighetsimplementering av de aritmetiske deler av en fraktal transformasjonsdekodingsmetode som vist på fig. 29. Følgelig vil det skjønnes at andre funksjoner som kunne utføres av en digital signalprosessor 612, ikke benyttes i den viste utførelse. Selv om den digitale signalprosessor 612 generelt opererer på 32 biter data, adderer den og multipliserer i den viste utførelse spesielt fortegnbestemt åtte biter data levert av minnegrensesnittet 610 som respons på en sekvens av instruksjoner inneholdt i programkoden ROM 602.

Disse primære komponenter dekker et sett av rekkevidde blokkadresser og overfører resultatene av dekodingsprosessen til skjermbufferen 604 og 605, som utgjør display minnebankene A og B og som i den viste utførelse består av en enkel 262 144 x 4 biter multiports video-RAM TMS44C251 fremstilt av Texas Instruments, Incorporated. Detaljer vedrørende virkemåten til de foretrukke displayminnebanker A og B er tilgjengelige i litteraturen levert av produsenten.

I den viste utførelse er datasettet som representerer et bilde en matrise av 256 x 256 piksler, selv om det vil skjønnes at størrelsen av denne matrisen er vilkårlig og at en større matrise gir opphav til større oppløsning. Dessuten skal det skjønnes at datasettet som representerer bildet, er det samme for skjermoppløsningen, skjermbufferoppløsningen og for alle aritmetiske

beregninger. I den viste utførelse er hver piksel-intensitetsverdi representert av en åtte biters verdi slik at bilder med 256 grå- eller pseudofarge intensitetsnivåer kan dekodes, selv om det vil skjønnes at størrelsen på denne matrisen er vilkårlig og at mer enn åtte biter pr. piksel gir opphav til mer enn 5 256 intensitets nivåer, mens mindre enn åtte biter pr. piksel gir opphav til mindre enn 256 intensitetsnivåer. Spesielt gir 24 biter pr. piksel opphav til tilstrekkelig intensitetsnivåer til å beskrive 256 intensitetsnivåer for hver av de røde, grønne og blå primærkomponenter.

10 De foretrukkede minnebanker A og B behandles begrepsmessig som to minnebanker selv om de fysisk er en innretning. En bank antas å inneholde "det forekommende bilde" og benyttes som kilden for billedata. Den andre anses å inneholde "det nye bilde" og benyttes som destinasjonen for bilde- data. Under den iterative dekodingsprosess, byttes minnebankenes rolle om.
15 Bankene skjelnes av en minnestart adresseforskyvning som for banken A er 0 og som for banken B er 131 076 (målt i firebiters minneposisjoner). Hver bank har tilstrekkelig størrelse til å romme et $256 \times 256 \times 8$ bits bilde. Minnegrensesnittet 610 inneholder adresse generatorer for å generere matriser av data svarende til kvadratiske underbilder. Underbildene bestem-
20 mes av en rad- og spalteadresse for det øvre venstre hjørne i dette under bilde og en størrelse. Underbildet med raden i øvre venstre hjørne lik R og spalte lik C og dimensjoner S er betegnt med sub (R,C,S). Sub (R,C,S) inneholder S ganger S piksel posisjoner og inneholder alle pikselposisjonene fra rad R til rad R+S-1 og fra spalte C+S-1.

25 Vertsgrensesnittet 600 mottar og sender signaler fra en ekspansjonsbuss 611 i en IBM PC/AT på PC-adresse-, -data- og busskontrollinjer 615. Data og adresser sendes til minne grensesnittet 610 på linjer 617. Minnegrensesnittet 610 består av minneadressegeneratorer og adressesekvensere. To typer av
30 data overfører minnegrensesnittet 610. Bildekoder plasseres på og hentes inn fra programdata-RAM 606 som er på 496×8 biter. Bildekoden kommer etter en databyte, format byten. De fire nedre biter, bitene 0, 1, 2, 3 av formatby- ten bestemmer domeneblokkstørrelsen og derfor rekkevidde blokkstørrelsen, da rekkeviddeblokkstørrelsen aldri er to ganger domeneblokkstørrelsen i den
35 viste utførelse. Gyldige verdier for domeneblokkstørrelsen er 4-16. Både domene blokkene og rekkeviddeblokkene er kvadratiske i den viste utførelse. En dimensjon spesifiserer således fullstendig domeneblokkstørrelsen.

- Formatbyten lagres i et statisk RAM-formatregister i minne grensesnittet 610. Formatbyten benyttes av den digitale signalprosessor 612 til å beregne de virkelige dimensjoner av de viste bildepikslar. Dimensjonen av det viste
- 5 bildet kan være noe mindre enn 256 x 256. Den bestemmes som det største antall pikslar som er et multippel av domeneblokkstørrelsen og som samtidig ikke er større enn 256. Hvis f.eks. domene blokkstørrelsen er 5, så er bildebredde og -høyde i pikslar 255 og ikke 256. Dette fordi 51 x 5 er 255, et tall som ikke er større enn 256, mens 52 x 5 er 260 som overstiger 256.
- 10 Antallet beregnes ved heltallsdivisjon av 256 med 5 og deretter med heltallsmultiplikasjon med 5. En liste av bildestørrelsesverdier som en funksjon av formatverdien er nå skaffet i formatet i, f, hvor i er bildestørrelsen og f er formatverdien:
- 15 256, 4; 255, 5; 252, 6; 252, 7; 256, 8; 252, 9; 250, 10; 253, 11; 252, 12; 247, 13; 252, 14; 255, 15; 256, 16.

Resultatet lagres i et statisk RAM-register med åtte biters minnestørrelse i minnegrensesnittet 610.

20

- En adressesekvenser i minnegrensesnittet 610 genererer med bruk av FORMAT (formatregisterverdien) adresser i program data-RAM 606 for å lagre bildekode som overføres fra vertsgrensesnittet 600 til minnegrensesnittet 610. Programdata RAM 606 består av 4096 minneposisjoner, hver
- 25 en byte bred. En bildekode kan bestå av opptil 2048 koder som hver opptar to byter av programdata RAM. CODELENGTH, antall to-byters programdata-RAM-ord som inneholder gyldige data, er kvadratet av verdien i bildestørrelse delt med verdien i FORMAT. CODELENGTH varierer mellom 256 og 2048 etter som verdien i format varierer mellom 4 og 16. De seksten bitene i
- 30 de første CODELENGTH ordene i programdata-RAM inneholder både adresse- og transformasjonsinformasjon. Delingen av hvert slikt ord mellom adresse- og transformasjonsinformasjonen avhenger av verdien i formatregisteret i henhold til den følgende formel:

35	FORMAT-verdi	Adressebiter	Transformasjons biter
	4-7	0-11	12-15
	8-15	0-9	10-15

16 0-7 8-15

Adressebitene deles i like deler for å danne en radadresse og en spalteadresse i henhold til den følgende tabell:

5	FORMAT-verdi	Radadressebiter	Spalteadressebiter
	4-7	0-5	6-11
	8-15	0-4	5-9
	16	0-3	4-7

- 10 Det vil skjønnes at selv om tallene for biter allokert til radadressene og spalteadressene er like i hvert kodeord, kan verdien av disse biter variere fra radadresse til spalte adresse og fra kodeord til kodeord. Radadressebitene og spalteadressebitene benyttes til å adresse øvre venstre posisjoner i rekkeviddeblokker. Gyldige verdier for rad- og spalteadressen varierer i henhold til verdien i FORMAT. Disse adressene kan variere fra 0 til en maksimumsverdi som uttrykt i den følgende liste. Listen er gitt som (m,f) hvor m er maksimumsadresseverdien for gitt formatverdi:

(63, 4) (50, 5) (41, 6) (35, 7) (31, 8) (27, 9) (24, 10) (22, 11) (20, 12) (18, 13) (17, 14)(16, 15) (15, 16)

- 20 I den viste utførelse blir pikselrad- og spalteadressene beregnet ved å multiplisere radadressen og spalteadresseverdiene med domeneblokkstørrelsen lagret som formatverdien. Adressegeneratoren i minnet 610 bestemmer deretter den fysiske minneadresse i displayminnebankene 604 og 605.

- 25 Transformasjonsbitene benyttes sammen med verdiene i NUMSCALE- og DENOMSCALE-registrene i minnegrensesnittet 610 for å utføre dekodingsoperasjonen med bruk av den fraktale gråskalatransformasjon. Den digitale signalprosessor 612 mottar begge de fortegnsbestemte åtte-biters
- 30 NUMSCALE- og DENOMSCALE-verdier over linjene 619. I den viste utførelse velges disse verdiene av en operatør og er fastlagt for en hel bildeberegning. Det skal skjønnes at anordningen av en større programdata-RAM ville tillate bruk av distinkte NUMSCALE- og DENOMSCALE-verdier for hvert kodeord. NUMSCALE- og DENOMSCALE-verdiene lastes fra
- 35 PC/AT-verten til verts grense snittene på linjene 615 og fra vertsgrensesnittet til NUMSCALE- og DENOMSCALE-registrene i minnegrensesnittet 610 på linjer 617.

Dekodingstrinnene i den fraktale gråskalatransformasjon utføres i et initialiseringstrinn og gjentatte dekodings iterasjonstrinn. De instruksjonene som er nødvendig for å utføre dekodingsprosedyren, lastes fra program-ROM 602
5 som er forbundet med den digitale signalprosessor 612.

Initialiseringstrinn: Et initialiseringssignal sendes fra PC/AT-verten på linjene 615 for å initialisere minnegrense snittet 610. Minnegrensesnittet 610 mottar nå 26 biter, hvorav de første åtte lagres i formatregisteret, de neste ni
10 lagres i NUMSCALE-registeret og de siste ni lagres i DENOMSCALE-registeret. En kodeteller blir nå initialisert til verdien 16 ganger CODELENGTH. Biter blir nå lastet fra PC/AT-verten over datalinjene 615 til vertsgrensesnittet og derfra over datalinjene 617 til minnegrensesnittet 610. En adressesekvenserer i minnegrensesnittet 610 initialisert av initialiserings-
15 signalet ovenfor, skaffer de påfølgende adresser i programdata-RAM 606 og overfører databitene den mottar til byteminneområdene som er slik adressert. Kodetelleren blir tellet ned med hver bit, og data overføringsprosessen stopper når den når 0. Et signal blir så sendt til den digitale signalprosessor 612 for å laste programmet fra programkode-ROM 602. Verdiene i
20 NUMSCALE- og DENOMSCALE-registrene overføres til minneposisjoner i RAM i den digitale signalprosessor 612. Verdien av FORMAT blir også overført til en minneposisjon i RAM i den digitale signalprosessor 612 som beregner og lagrer verdien av CODELENGTH som beskrevet ovenfor. Det annet adressested, displaybanken, i den digitale signalprosessor RAM slettes.
25

Dekodingsiterasjonstrinn: En minneposisjon konfigurert som en blokketeller i RAM i den digitale signalprosessor 612 innstilles på startadressen til programdata-RAM 606 som er inneholdt i programkode-ROM 602. Hvert dekodings iterasjonstrinn består av CODELENGTH-iterasjoner av blokk
30 dekodningstrinn. Verdien av displaybanken sendes til minnegrensesnittet 610 og lagres i et bankregister i minnegrensesnittet 610.

Blokkdekodingstrinn: Ved starten av et blokkdekodingstrinn blir verdien av BLOCK COUNTER i den digitale signalprosessor 612 levert på datalinjene
35 619 til minnegrensesnittet 610. De åtte bitene av data på denne adresse i programdata-RAM og de neste åtte bitene av data leveres tilbake til minnegrensesnittet 610 på datalinjen 619. Transformasjonsbitene for kodeordet

sendes på datalinjene 13 til den digitale signalprosessor 612 og lagres i et minnested Q i den digitale signalprosessor 612. Rad- og spalteadressebitene til kodeordet benyttes av displayadressegeneratoren på minne grensesnittet 610 for å generere adressene til en kvadratisk blokk med horisontal og vertikal dimensjon lik to ganger FORMAT, av pikseldata i displayminnebanken. Hvis verdien av displaybanken er null, så blir adressene generert i displayminnebanken ved 605. Det øvre venstre hjørne i den kvadratiske blokk genereres som dette pikselet på FORMAT ganger radadressen ROW og FORMAT-adressen ganger spalte adressen COLUMN. Bildedataene i denne kvadratiske blokk er SUB (ROW ADDRESS x FORMAT,COLUMN ADDRESS x FORMAT, 2 x FORMAT). Adressene blir generert som 2 ganger 2 underblokker av denne kvadratiske blokk, det vil si med radadresser SUBROW - ROW ADDRESS x FORMAT + 2 x i og spalteadresser SUBCOL = COLUMN ADDRESS x + 2 x j.

Adressegeneratoren initialiserer tellere med verdier i og j med 0 og øker deretter j inntil j når FORMAT, på hvilket tidspunkt den tilbakestilles til 0. Når j er tilbakestilt, så blir i øket inntil i når FORMAT, hvor den tilbakestilles til 0. Et register i den digitale signalprosessor 612, PIXEL, blir slettet. For hver verdi av i og j mellom 0 og FORMAT-1 inklusive, blir byteverdiene ved byten av piksel data lagret ved adresseposisjoner for fire piksler i SUB (SUBROW, SUBCOL, 2), sendt over adresselinjene 619 til minnegrensesnittet 610 og deretter over datalinjene 619 til den digitale signalprosessor 612 og akkumulert i PIXEL-registeret.

Etter at alle fire verdier er blitt mottatt, blir verdien forskjøvet to plasser til høyre for å frembringe en deling med 4. Q blir addert til resultatet av (PIXEL x NUMSCALE dividert med DENOMSCALE), og resultatet av denne beregningen lagres i et digitalt signalprosessorregister 612, NEWPIXEL.

En annen adressegenerator i minnegrensesnittet 610 bestemmer destinasjonsadresser i displayminnet. Pikselkoordinatene til koordinatene i det øvre venstre hjørne beregnes av den annen adressegenerator. Pikselraden for denne posisjon er

$$\text{DESTROM} = (\text{BLOCKCOUNT}/(\text{IMAGE SIZE}/\text{FORMAT})) \text{ ganger } \text{FORMAT}$$

Pikselspalteadressen i denne posisjon er

5 $DESTCOL = (BLOCKCOUNT\%(IMAGESIZE/FORMAT)) \text{ ganger}$
 $FORMAT$

hvor $X\%Y$ angir resten av heltallsdivisjonen av X med Y .

Adressen til $FORMAT$ ganger $FORMAT$ -pikslene i

10

$SUB (DESTROW, DESTCOL, FORMAT)$

15 som beregnet i tellerne i og j , varierer. $NEWPIXEL$ sendes over datalinjer 13 til minnegrensesnittet 610 og lagres i destinasjonsdisplayminneadressen for pikselet som befinner seg i $rad = DESTROW + i$ og spalte $DESTCOL + j$. Når alle $FORMAT$ ganger $FORMAT$ -piksler i $SUB (DESTROW, DESTCOL, FORMAT)$ er skiftet ut med $NEWPIXEL$ -verdier på denne måte, blir tellerne i og j slettet og $BLOCKCOUNT$ øker.

20 Dette avslutter et blokkdekodingstrinn. Blokk dekodingstrinnet er gjentatt $CODELENGTH$ mange ganger med de økede blokkteilverdier. Verdien av displaybanken erstattes av 1 - verdien av displaybanken og veksler følgelig mellom 0 og 1. Dekodingsiterasjonstrinnet er dermed fullført.

25 Visningen av bildet utføres essensielt asynkront med den beregningsmessige dekoding. Pikselerverdiene blir faktisk utlest av displayminnet asynkront, men minnebanken som benyttes som kilde for displaypikslene spesifiseres av verdien av displaybanken. Displayutgangsgrensesnittet 613 gir R-G-B-utgangen. Den logiske taktinnretning 617 på video utgangen driver den foretrukkede fargepalett, en Brooktree Bt453RAMDAC, ikke vist på fig. 31.

30

Forholdet mellom bildekodestørrelsen og den dekodete bilde størrelse er kalt kompresjonsforholdet til innretningen. Kompresjonsforholdet til innretningen kan beregnes ved å dele to ganger $CODELENGTH$ med kvadratet av bildestørrelsen. Den viste utførelse gir kompresjonsforhold som varierer fra 35 1:16 når $FORMAT$ inneholder domeneblokkstørrelsesverdien på 4 og til et kompresjonsforhold på 1:128 når $FORMAT$ -verdien er 16. Begrensningen av

FORMAT-verdier til å variere mellom 4 og 16 er vilkårlig. Større FORMAT-verdier gir større kompresjonsforhold.

- 5 Et tiltenkt formål med den viste dekodingsinnretning er å skaffe et økonomisk middel til å dekode bilder som er blitt kodet med den fraktale gråskala-transformasjon. Nytten av innretningen kan altså måles uttrykt ved det kompresjonsforhold den oppnår. Kompresjonsforholdet kan effektivt økes ved å komprimere de fraktale transformasjonskoder med tradisjonell kompresjon av en fraktal gråskalatransformasjon. F.eks. kunne offentlig
- 10 tilgjengelige huffman-kodeskjemaer benyttes til å kode de fraktale transformasjonskoder. Huffman-kodede fraktale transformasjonskoder blir da huffman-dekodet før overføringen til det viste apparat for å dekode den fraktale transformasjon.
- 15 Bruk av de fraktale transformasjonsmetoder som er beskrevet her, har en vesentlig fordel fremfor kjente metoder såsom vektorkvantisering og hufmann-koding i og med at oppslags tabeller ikke benyttes. Således kan kompresjonsforholdet lett forandres og systemet benyttes i en adaptiv mode som er nyttig i anvendelser såsom DVI og behandling av gamle filmer for
- 20 overføring på telefonlinjer.

- De viste metoder og det viste apparat er spesielt egnet for anvendelse på problemet med overføring av filmer over standard telefonlinjer. Fig. 32 viser et logisk flytkart for en fremgangsmåte til å generere komprimerte digitale
- 25 data som representerer bildeinformasjon i et filmbilde ved hjelp av en dataprosessor, for lagring eller overføring. I trinn 702 blir første og andre behandlingsbufferne initialisert med forhåndsbestemte bildedata. I den foretrukkede utførelse blir behandlingsbufferne initialisert til verdier på null. Fortrinnsvis blir behandlingsbufferne dannet for å lagre gråskalapikselverdier
- 30 av en kvadratisk bildematrise. I en foretrukket utførelse skaffer behandlingsbufferne lagring av en 256 x 256 matrise av 8-biters piksler. Verdien av telleren i dataprosessoren blir også initialisert i trinn 702.

- I trinn 704 blir dernest bildeinformasjonen inneholdt i det første enkeltbilde
- 35 av filmen som skal behandles, digitalisert for å danne de digitaliserte bildedata. Bildeinformasjonen kan digitaliseres ved samme oppløsning som behandlingsbufferne. I den foretrukkede utførelse blir bildeinformasjon fra

det første enkeltbilde på filmen digitalisert til en 256 x 256 matrise av piksler med en 8-biters gråskalaverdi.

5 I trinn 706 blir digitaliserte bilde­data fra det første enkeltbilde på filmen lagret i en bildebuffer. Tellerverdien blir deretter sjekket for å bestemme hvorvidt verdien er større enn en forhåndsbestemt grenseverdi. Telleren benyttes med det formål å gjenta behandlingen av den første ramme et fast antall ganger, f.eks. åtte. Dvs. at grenseverdien referert til i trinn 708 settes til 8. Hvis tellerverdien er større enn grenseverdien, blir informasjon fra det 10 neste sekvensielle enkeltbilde i filmen digitalisert for å frembringe de digitaliserte bilde­data for det neste enkeltbilde, og de digitaliserte bilde­data for det neste enkeltbilde blir lagret i en bildebuffer. Hvis telleverdien ikke er større enn grenseverdien, som bestemt i trinn 708, overhoppes trinn 710 med det resultat at bilde­data fra det første enkeltbilde forblir i enkeltbildebufferen. 15

I trinn 712 blir bilde­data fra enkeltbildebufferen lagret i den første behandlingsbuffer. Disse bilde­data blir deretter behandlet på samme måte som vist på fig. 5. Spesielt dannes domeneblokker i trinn 714. I trinn 716 blir dernest 20 krympede rekkeviddeblokker dannet i den annen behandlingsbuffer. For hver domeneblokk blir den nærmeste krympede rekkeviddeblokk valgt i trinn 718 og den valgte krympede rekkeviddeblokk benyttes til å erstatte den tilsvarende domeneblokk i den første behandlingsbuffer i trinn 720.

25 Sett­et av identifika­torer for de valgte krympede rekkeviddeblokker blir deretter benyttet i trinn 722 for å danne en fraktaltransformasjon såsom fraktaltransformasjon 410, vist på fig. 26. Denne fraktaltransformasjon er betegnet som en forekommende transformasjon. Den nåværende transformasjon som omfatter komprimerte digitale data som representerer bildeinformasjonen i det første enkeltbilde i filmen, blir deretter levert som utdata i 30 trinn 724, og innholdet i den første behandlingsbuffer kopiert på den annen behandlings buffer i trinn 726.

35 I trinn 728 blir tellerverdien øket, og det gjøres en bestemmelse i blokk 730 om hvorvidt alle enkeltbilder av filmen er blitt behandlet. Hvis ikke, vender behandlingen tilbake til trinn 708. Hvis alle delbilder er blitt behandlet, avsluttes fremgangsmåten ved 732.

På fig. 33 er det vist et logisk flytkart for en fremgangsmåte til automatisk generering av digitale data som representerer delbildene i en film. Dataene genereres av en dataprosessor fra en sekvens av nåværende transformasjoner som hver har et sett av identifikatorer. Dataprosessoren innbefatter et minne med en rekke buffere som hver har en rekke adresserbare områder spesifisert av en bufferadresse. Hver av identifikatorene svarer til et adresserbart område i bufferne og innbefatter en bufferadresse og en prosedyrespesifikasjon. I den foretrukkede utførelse er det skaffet et par av buffere A og B som i størrelse svarer til behandlingsbufferne omtalt i tilknytning til fig. 32.

Spesifikt har bufferne A og B i den foretrukkede utførelse hver en størrelse tilstrekkelig til å lagre en matrise på $256 \times 256 \times 8$ biter.

I trinn 800 utføres et initialiseringstrinn hvorved et forhåndsbestemt mønster av digitale data lagres på minst en av bufferne, bufferen B betegnet som en kildebuffer. Fortrinnsvis blir begge bufferne A og B initialisert til bare nuller. En nåværende transformasjon, fortrinnsvis generert i en prosess som vist på fig. 32, legges inn i trinn 802 fra sekvensen av nåværende transformasjoner som utgjør komprimerte digitale data som representerer en film. I trinn 804 blir de digitale data i buffer A omdannet til domeneblokker som hver svarer til en identifikator i den nåværende inn gangstransformasjon. I trinn 806 blir de digitale data i buffer B omdannet til krympede rekkeviddeblokker på samme måte som omtalt ovenfor i tilknytning til fig. 13. I trinn 808 blir hver domeneblokk i buffer A erstattet av en krympet rekkeviddeblokk. Mer bestemt fastsettes for hver av identifikatorene i den nåværende transformasjon som er lagt inn i trinn 802, et mønster av data som svarer til hver av identifikatorene. Dette datamønster bestemmes ved å benytte prosedyrene angitt i denne identifikator på delen av kildebufferen, buffer B, som angis ved bufferadressen i identifikatoren. I trinn 808 blir dernest dataene lagret i de adresserbare områder i buffer A, betegnet som en destinasjonsbuffer, og som angitt ved adressene i de tilsvarende identifikatorer. I den foretrukkede utførelse blir hver domeneblokk i buffer A erstattet av en krympet rekke viddeblokk i buffer B. Denne prosessen er identisk med den beskrevet ovenfor i tilknytning til fig. 13.

I trinn 810 blir så innholdet i destinasjonsbufferen levert som data som representerer et bilde i enkeltbildet. Dvs. at bildedata fra buffer A gis ut som digitale data som representerer et bilde i enkeltbildet. Dette bildet svarer til

enkeltbildet som er representert ved den nåværende transformasjon som er lagt inn i trinn 802.

5 I trinn 812 gjøres det en bestemmelse om hvorvidt alle nåværende transformasjoner er blitt behandlet, dvs. hvorvidt bilder som svarer til alle ønskede enkeltbilder er blitt behandlet. Hvis dette er tilfellet, opphører prosessen i trinn 814. Ellers vil innholdet i destinasjonsbufferen kopieres på kildebufferen. Det vil si at innholdet i buffer A som vist i trinn 816, kopieres på buffer B, og behandlingen vender tilbake til trinn 802 for å dekomprimere den neste 10 nåværende transformasjon av sekvensen. Sekvensen av sett av billedata som gis ut i trinn 810, utgjør de dekomprimerte billedata som representerer hvert enkeltbilde i en film.

15 Prinsippene for den foreliggende oppfinnelse er også spesielt egnet til bruk på en videofon. Fig. 34 viser et blokk diagram av et apparat 1000 for å tilveiebringe en videofon. Apparatet 1000 omfatter to lås-og-hold inngangsenheter 1011 og 1012, en kontrollogikkenhet 1013, to dataprosessorer 1014 og 1015 (som henholdsvis fungerer som en videofonkoder og en videofondekoder) og to utgangsenheter 1017. Apparatet 1000 har to funksjonelle moder, 20 koding og dekoding. Kontrollogikkenheten 1013 velger enten kodingsmoden eller dekodingsmoden som bestemt ved inngangsdata.

I kodingsmoden skaffer en videokilde 1002, så som et videokamera et analogt signal til en enkeltbildegripekrets 1006 som omdanner hvert enkeltbilde av videoen fra kilden 1002 til digitale data og lagrer de digitale data i 25 enkeltbildegripekretsen 1006. Kontrollogikkenheten 1013 får video inngangsenheten 1011 til å lese de digitale data lagret i enkeltbildegripekretsen 1006 og fører dataene videre til videofonenkoderen 1014. Videofonkoderen 1014 er en fraktal koder av typen som er beskrevet tidligere i tilknytning til 30 fig. 6. Videofonkoderen 1014 behandler de digitale videodata som den mottar fra kontrollogikkenheten 1013. Denne behandlingen frembringer fraktaltransformasjoner for enkeltbildene som deretter leveres ut gjennom datautgangsenheten 1016. Datautgangsenheten 1016 kan være en standard telefon modemkrets.

35 I dekodingsmoden blir fraktale transformasjoner levert av en datakilde 1004 som også kan være en standard telefon modemkrets. Kontrollogikkenheten

- 1013 får data inngangsenheten 1012 til å lese og akkumulere dataene levert av datakilden 1004 og føre den til videofondekoderen 1015. Videofondekoderen 1015 er en fraktal dekker av den art som ble beskrevet i tilknytning til fig. 14. Videofondekoderen 1015 behandler de fraktale transformasjoner som mottas fra kontrollogikken 1013. Behandlingen resulterer i digitale videodata som sendes fra dekkeren 1015 til video utgangsenheten 1017. Denne utgangsenheten omformer de digitale data til analoge data for visning på videodisplayenheten 1020.
- 10 Den foreliggende oppfinnelse skaffer datakompresjonsmetoder og et apparat som har høyere ytelse enn kjente metoder og apparater. Dessuten unngår den foreliggende oppfinnelse nødvendigheten av timer med manuelt arbeid på en datamaskin for å oppnå komprimerte data. Da oppslagstabeller ikke er nødvendige, kan den foreliggende oppfinnelse gi adaptive muligheter
- 15 samtidig som mengden av minne som er nødvendig, minimeres.
- Oppfinnelsen skaffer muligheten for å representere og lagre bilder i en digital datamaskin med en skalauavhengig oppløsning ved bruk av en todimensjonal matrise av numeriske attributter for et digitalisert bilde som inndata og
- 20 skaffer en utdata som er et mindre datasett, fra originalen. Den foreliggende oppfinnelse kan både gi eksakt og ikke-eksakt bilderepresentasjon ved valg av passende blokkstørrelser og skjermopløsning.
- Oppfinnelsen skaffer videre evnen til å utsette bildedata for en rekke
- 25 passasjer gjennom en kodings- og dekodings prosess slik at bildedataene kommer ut som de samme, som om de hadde blitt kodet og dekodet bare en gang.

PATENTKRAV

1. Fremgangsmåte til automatisk behandling av digitale bilde­data bestående av bildeinformasjon, hvor fremgangsmåten er
k a r a k t e r i s e r t v e d at den omfatter følgende trinn utført med en
5 dataprosessor:
 å lagre bilde­dataene i dataprosessoren,
 å generere en rekke entydig adresserbare domeneblokker fra de lagrede bilde­data, idet hver av domeneblokkene representerer en forskjellig del av bildeinformasjonen slik at all bildeinformasjon er inneholdt i minst en av
10 domeneblokkene,
 å danne fra de lagrede bilde­data en rekke entydig adresserbare, avbildete rekkeviddeblokker, svarende til forskjellige undermengder av lagrede bilde­data og med en entydig adresse til hver av undermengdene, idet dannelses-
15 trinnet innbefatter et undertrinn for å eksekvere for hver av de avbildete rekkeviddeblokker en krympeprosedyre på undermengden av lagrede bilde­data som svarer til den avbildete rekkeviddeblokk,
 å tilordne entydige identifikatorer til tilsvarende avbildete rekkeviddeblokker, idet hver av identifikatorene for den tilsvarende avbildete rekkeviddeblokk spesifiserer en prosedyre og en adresse for den tilsvarende under-
20 mengde av lagrede bilde­data,
 å velge for hver av domeneblokkene den av de avbildete rekkeviddeblokker som mest nøyaktig svarer til den respektive domeneblokk i henhold til forhåndsbestemte kriterier, og
25 å gjengi bildeinformasjonen som et sett av identifikatorer for de valgte, avbildete rekkeviddeblokker.
2. Fremgangsmåte i henhold til krav 1,
k a r a k t e r i s e r t v e d at de digitale bilde­data består av en rekke
30 piksler, og at trinnet for å generere domeneblokkene innbefatter et undertrinn for å generere domeneblokkene med ulike antall piksler.
3. Fremgangsmåte i henhold til krav 1,
k a r a k t e r i s e r t v e d at de digitale bilde­data består av en rekke
35 piksler, og at trinnet for å generere domeneblokkene innbefatter et undertrinn for å generere hver av domeneblokkene med det samme antall piksler.

4. Fremgangsmåte i henhold til krav 1-3,

karakterisert ved at trinnet for å generere en rekke entydige adresserbare, avbildete rekkeviddeblokker innbefatter undertrinn for

5 å danne en rekke initiale rekkeviddeblokker som har større antall piksler enn domeneblokkene,

å krympe de initiale rekkeviddeblokker for å frembringe krympede rekkeviddeblokker som har samme antall piksler som domeneblokkene, og

10 å benytte de krympede rekkeviddeblokker til å danne de avbildete rekkeviddeblokker.

5. Fremgangsmåte i henhold til krav 4,

karakterisert ved at krympetrinnet for krymping omfatter et undertrinn for å velge bestemte piksler i hver initial rekkeviddeblokk for å
15 danne en tilsvarende krympet rekkeviddeblokk.

6. Fremgangsmåte i henhold til krav 5,

karakterisert ved at undertrinnet for å velge piksler innbefatter et undertrinn for å velge piksler som har samme innbyrdes avstand i de initiale
20 rekkeviddeblokker.

7. Fremgangsmåte i henhold til krav 4,

karakterisert ved at krympetrinnet innbefatter et undertrinn for å midle tilstøtende piksler.

8. Fremgangsmåte i henhold til et av kravene 1-7,

karakterisert ved at trinnene for å generere en rekke entydig adresserbare, avbildete rekkeviddeblokker omfatter et undertrinn for å
30 benytte en affin transformasjon på en av undermengdene av det lagrede bilde som svarer til den avbildete rekkeviddeblokk, og at trinnet for å tilordne entydige identifikatorer omfatter et undertrinn for å tilordne entydige identifikatorer som hver innbefatter både en adresse og en spesifikasjon av en affin transformasjon.

35 9. Fremgangsmåte i henhold til krav 8,

karakterisert ved at trinnet for å danne en rekke entydig adresserbare, avbildete rekkeviddeblokker innbefatter et undertrinn for å invertere den avbildete rekkeviddeblokk.

- 5 10. Fremgangsmåte i henhold til krav 1-9,
karakterisert ved at trinnet for å generere domeneblokkene
innbefatter et undertrinn for å tilordne hvert parti av den lagrede bildeinformasjon til bare en av domeneblokkene slik at domeneblokkene ikke er
overlappende.

10

11. Fremgangsmåte i henhold til krav 1-10,
karakterisert ved at trinnet for å velge den av de avbildete
rekkeviddeblokker som stemmer best overens, innbefatter et trinn for å velge
for hver domene-blokk den avbildete rekkeviddeblokk som har den minimale
15 hausdorff-avstand fra domeneblokken.

12. Fremgangsmåte i henhold til krav 1-10,
karakterisert ved at trinnet for å velge den best samsvarende av
de avbildete rekkeviddeblokkene innbefatter et trinn for å velge for hver
20 domeneblokk den avbildete rekkeviddeblokk med den minimale modifiserte
hausdorff-avstand beregnet som $D[db, mrb] + D[1-db, 1-mrb]$, hvor D er en
avstand beregnet mellom et par av mengder av data som er representert i et
bilde, db er en domeneblokk, mrb er en avbildet rekkeviddeblokk, 1-db er
den inverse av en domeneblokk og 1-mrb er den inverse av en avbildet
25 rekkeviddeblokk.

13. Fremgangsmåte i henhold til krav 1-10,
karakterisert ved at trinnet for å velge den best samsvarende av
de avbildete rekkeviddeblokker innbefatter undertrinn for å:
30 beregne for hver av domeneblokkene den veiede L^2 -avstand D_L fra hver
av de avbildete rekkeviddeblokker som

$$D_L = \sqrt{\sum_{i=1}^N |x_i - y_i|^2 w_i}$$

35

hvor x_i , y_i er indekserte intensitetsverdier i de respektive bilder, w_i er ikke-negative tall og N er antall piksler i hver domeneblokk, og å velge for hver domeneblokk den avbildete rekkeviddeblokk med den minimale veiede L^2 -avstand.

5

14. Fremgangsmåte i henhold til krav 1-13,
karakterisert ved at trinnet for å danne avbildete rekkevidde-
blokker innbefatter et trinn for å danne en avbildet rekkeviddeblokk som
omfatter en utbryterblokk som har den samme størrelse som de avbildete
10 rekkeviddeblokker og inneholder digitale data som representerer et helhvitt
bilde.

15. Fremgangsmåte i henhold til et av kravene 1-14,
karakterisert ved at de digitale bildedata omfatter en rekke
15 piksler som hver har en rekke gråskalaverdier.

16. Fremgangsmåte i henhold til krav 15,
karakterisert ved at trinnet for å velge innbefatter undertrinn for
å omforme gråskalaverdiene av pikslene i hver domeneblokk med en
20 lineær funksjon som innbefatter minst en vertikal skaleringsparameter,
å omforme gråskalaverdiene av pikslene i de krympede rekkeviddeblok-
ker ved hjelp av den lineære funksjon,
å velge for hver av domeneblokkene den av de avbildete rekkevidde-
blokker som har omformede pikselverdier som best stemmer overens med de
25 omformede pikselverdier for domeneblokken i henhold til de første for-
håndsbestemte kriterier,

og hvor trinnet for å representere bildeinformasjon innbefatter et
undertrinn for å representere bildeinformasjonen som en mengde av verdier
som hver inkluderer en identifikator for den valgte avbildete rekkevidde-
30 blokk og minst en vertikal skaleringsparameter.

17. Fremgangsmåte i henhold til krav 16,
karakterisert ved at det første kriterium innbefatter minimering av
hausdorff-avstanden mellom hver domeneblokk og den valgte, krympede
35 rekkeviddeblokk.

18. Fremgangsmåte i henhold til krav 16,

5 k a r a k t e r i s e r t v e d at undertrinnene for å omforme pikselverdier
for domeneblokkene og krympede rekkeviddeblokker omfatter å velge minst
en vertikal skaleringsparameter for å maksimere overenstemmelsen mellom
domeneblokkene og de krympede rekkeviddeblokker i henhold til et annet
-forhåndsbestemt kriterium.

10 19. Fremgangsmåte i henhold til krav 18,
k a r a k t e r i s e r t v e d at de første og andre forhåndsbestemt kriterier
er forskjellige.

15 20. Fremgangsmåte i henhold til krav 19,
k a r a k t e r i s e r t v e d at det annet forhåndsbestemte kriterium
innbefatter minimering av L^2 -avstanden mellom domeneblokkene og de
krympede rekkeviddeblokker.

20 21. Fremgangsmåte i henhold til et av kravene 1-20, hvor de digitale data
representerer en film som har en rekke individuelle, sekvensielle enkeltbilder
som hver inneholder enkeltbildeinformasjon,
k a r a k t e r i s e r t v e d at den omfatter trinn utført av en dataprosessor
for

25 å initialisere første og andre behandlingsbuffer i dataprosessoren,
 å lagre de digitale bildedata i den første behandlingsbuffer,
 å utføre trinnet for å generere en rekke entydig adresserbare domene-
blokker ved å danne i den første behandlingsbuffer en rekke entydig adres-
30 serbar domeneblokker fra de lagrede bildedata i den første behandlingsbuffer,
idet hver av domeneblokkene representerer et forskjellig parti av enkeltbilde-
informasjonen slik at all enkeltbildeinformasjon er inneholdt i minst en av
domeneblokkene,

35 å utføre trinnet for å danne en rekke entydig adresserbare, avbildete
rekkeviddeblokker ved å danne fra data lagret i den annen behandlingsbuffer
en rekke entydig adresserbare, avbildete rekkeviddeblokker som svarer til
forskjellige undermengder av de lagrede data i den annen behandlingsbuffer,
idet hver av undermengdene har en entydig adresse, og idet dannelsesstrinnet
innbefatter undertrinnet å eksekvere for hver av de avbildete rekkevidde-
blokker en tilsvarende prosedyre på en av undermengdene av data i den
annen behandlingsbuffer og som svarer til den avbildete rekkeviddeblokk,

å utføre trinnet for å tilordne entydig identifikatorer ved å tilordne entydige identifikatorer til tilsvarende i de avbildete rekkeviddeblokker, idet hver av identifikatorene spesifiserer for den tilsvarende avbildete rekkeviddeblokk en prosedyre og en adresse for den tilsvarende undermengde av lagrede data i den annen behandlingsbuffer,

å utføre trinnet for selektering ved å selekttere for hver av domeneblokkene i den første behandlingsbuffer den av de avbildete rekkeviddeblokker i den annen behandlingsbuffer som best stemmer overens i henhold til forhåndsbestemte kriterier,

å utføre det ytterligere trinn å erstatte hver av domeneblokkene med den tilsvarende valgte, avbildete rekkeviddeblokk,

å utføre trinnet for å representere bildeinformasjonen ved å danne en forekommende transformasjon som et sett av identifikatorene for de avbildete rekkeviddeblokker,

å utføre ytterligere trinn for å kopiere innholdet i den første behandlingsbuffer på den annen behandlingsbuffer,

å gjenta trinnene for lagring, dannelsen av den første og annen behandlingsbuffer, tilordning, selektering, utskifting, dannelsen og kopiering inntill alle enkeltbildene er blitt digitalisert, og

å levere sekvensen av forekommende transformasjoner for hver av enkeltbildene som en komprimert representasjon av bildeinformasjonen i filmen.

22. Fremgangsmåte i henhold til krav 21,

karakterisert ved at bildedataene fra det første enkeltbilde blir gjentatt behandlet et forhåndsbestemt antall ganger for å danne en serie av forekommende transformasjoner som er representative for det første enkeltbilde, og å behandle bildedata fra hvert påfølgende enkeltbilde én gang for å danne en enkelt forekommende transformasjon som er representativ for hvert påfølgende enkeltbilde.

23. Fremgangsmåte til automatisk generering av digitale data som representerer enkeltbildene i en film ved hjelp av en dataprosessor fra sekvensen av forekommende transformasjoner generert ved fremgangsmåten i henhold til krav 21, hvor hver av transformasjonene har en identifikator som omfatter komprimerte digitale data som representerer enkeltbildene, hvor dataprosessoren innbefatter et minne med flere buffere som hver har en rekke adresser-

bare områder spesifisert av en bufferadresse, og hvor hver av identifikatorene svarer til et adresserbart område i bufferen og innbefatter en bufferadresse og en prosedyrespesifikasjon, og hvor fremgangsmåten er

5 k a r a k t e r i s e r t v e d at den omfatter trinn utført av dataprosessoren for

 å initialisere et forhåndsbestemt mønster av digitale data inn i en av bufferne utpekt som en kildebuffer,

 å legge inn en forekommende transformasjon og lagre den forekom-
10 mende transformasjonen i en annen av bufferne betegnet som destinasjons-
 bufferen,

 å bestemme for hver av identifikatorene et mønster av data svarende til hver identifikator ved å benytte prosedyren spesifisert i denne identifikator på partiet av kildebufferen angitt av bufferadressen i denne identifikator,

 å lagre mønstrene av data i de adresserbare områder av destinasjonsbuf-
15 feren og som er spesifisert av de tilsvarende identifikatorer,

 å levere innholdet i destinasjonsbufferen som data representative for en bildeinformasjon i et enkeltbilde,

 å kopiere innholdet i destinasjonsbufferen over på kildebufferen,
 og å gjenta trinnene for innlegging, bestemmelse, lagring, levering og
20 kopiering for hver forekommende transformasjon av sekvensen.

24. Fremgangsmåte til å generere et bilde ved hjelp av en dataprosessor for et sett av identifikatorer frembrakt ved fremgangsmåten i henhold til krav 1, hvor dataprosessoren innbefatter et minne som har en buffer med en rekke
25 adresserbare områder spesifisert av en bufferadresse, og hvor hver av
 identifikatorene svarer til et adresserbart område i bufferne og innbefatter en
 bufferadresse

 k a r a k t e r i s e r t v e d at fremgangsmåten omfatter følgende av en dataprosessor utførte trinn for

30 å lagre et forhåndsbestemt mønster av digitale data i bufferen,

 å bestemme for hver av identifikatorene et mønster av data svarende til hver identifikator ved å velge data i et parti av bufferen som er angitt ved bufferadressen i denne identifikator,

 å lagre mønstrene av data i adresserbare områder svarende til identifika-
35 torene,

 å gjenta trinnene for bestemmelse og lagring i bufferen inntil forhånds-
 bestemte kriterier er oppfylt, og

å levere innholdet av bufferen som data representative for et bilde når de forhåndsbestemte kriterier er oppfylt.

25. Fremgangsmåte i henhold til krav 24,

5 k a r a k t e r i s e r t v e d at de digitale bildedata består av bildeinformasjon med en første oppløsning, og at utgangsbildedataene er representative for inngangsbildet ved en annen oppløsning som er høyere enn den første oppløsning.

10 26. Fremgangsmåte i henhold til krav 24, hvor dataprosessoren innbefatter et minne med en rekke buffere,

k a r a k t e r i s e r t v e d at trinnet for lagre et forhåndsbestemt mønster av digitale data omfatter

15 å lagre et forhåndsbestemt mønster av data i en av bufferne betegnet som en kildebuffer,

at trinnet for å bestemme et mønster av data omfatter å bestemme for hver av identifikatorene et mønster av data svarende til hver identifikator ved partiet av kildebufferen angitt av bufferadressen i denne identifikator,

20 at trinnet for å lagre mønstrene av data omfatter å lagre mønstrene av data i det adresserbare område i en annen av bufferne betegnet som en destinasjonsbuffer, svarende til identifikatoren,

at trinnet for å gjenta trinnene for bestemmelse og lagring i bufferen omfatter å gjenta trinnene for bestemmelse og lagring i destinasjonsbufferen, idet destinasjonsbufferen betraktes som kildebufferen, inntil forhåndsbestemte kriterier er oppfylt,

25 og at trinnet for å levere innholdet i bufferen omfatter å skaffe innholdet i destinasjonsbufferen som data representative for et bilde når de forhåndsbestemte kriterier er oppfylt.

30 27. Fremgangsmåte i henhold til krav 26, hvor identifikatoren omfatter både en bufferadresse og en prosedyrespesifikasjon,

k a r a k t e r i s e r t v e d at trinnet for å bestemme et mønster av data omfatter å bestemme for hver av identifikatorene et mønster av data som svarer til hver identifikator ved å benytte prosedyren spesifisert av denne identifikator på partiet av kildebufferen angitt av bufferadressen i denne identifikator.

35

28. Fremgangsmåte i henhold til krav 26,
karakterisert ved at trinnet for å levere innholdet i destinasjons-
bufferen når de forhåndsbestemte kriterier er oppfylt, innbefatter et under-
trinn for å bestemme når innholdet i kilde- og destinasjonsbufferne skiller
5 seg med mindre enn en forhåndsbestemt verdi.

29. Fremgangsmåte i henhold til krav 26,
karakterisert ved at trinnet for å levere innholdet i destinasjons-
bufferen som data representative for et bilde når de forhåndsbestemte
10 kriterier er oppfylt, innbefatter trinnene for å bestemme og lagre i destina-
sjonsbufferen et forhåndsbestemt antall ganger.

30. Fremgangsmåte i henhold til krav 26,
karakterisert ved at trinnene for å lagre digitale data i destina-
15 sjonsbufferen innbefatter et trinn for å lagre minst et av mønstrene i hvert
adresserbart område i destinasjonsbufferen.

31. Apparat for automatisk å behandle digitale bilde data bestående av
bildeinformasjon, hvor apparatet omfatter en anordning for å lagre bilde data,
20 en anordning for å generere en rekke entydige adresserbare domeneblokker
(34) fra de lagrede bilde data, hvor hver av domeneblokkene representerer et
forskjellig parti av bildeinformasjonen slik at all bildeinformasjon er innholdt
i minst en av domeneblokkene, en anordning for å danne fra de lagrede
bilde data en rekke entydig adresserbare, avbildete rekkeviddeblokker
25 (36, 38, 40) svarende til forskjellige undermengder av de lagrede bilde data
og med en entydig adresse for hver av undermengdene, og hvor dannelsesan-
ordningen innbefatter en anordning for å eksekvere for hver av de avbildete
rekkeviddeblokker en krympeprosedyre på en av undermengdene av de
lagrede bilde data som svarer til den avbildete rekkeviddeblokk,
30 karakterisert ved en anordning for å tilordne entydige identifika-
torer til de hver av de avbildete rekkeviddeblokker, idet hver av identifika-
torene for den tilsvarende avbildete rekkeviddeblokk spesifiserer en adresse for
den tilsvarende undermengde av lagrede bilde data,
en anordning for å velge for hver av domeneblokkene den av de
35 avbildete rekkeviddeblokker som best stemmer overens i henhold til for-
håndsbestemte kriterier,

og en anordning for å representere bildeinformasjon som en mengde av identifikatorer for valgte, avbildete rekkeviddeblokker.

32. Apparat i henhold til krav 31,

5 k a r a k t e r i s e r t v e d at anordningen for å tilordne entydige identifikatorer omfatter en anordning for å tilordne entydige identifikatorer til tilsvarende avbildete rekkeviddeblokker, idet hver av identifikatorene for den tilsvarende avbildete rekkeviddeblokk spesifiserer en prosedyre og en adresse for den tilsvarende undermengde av lagrede billedata.

10

33. Apparat i henhold til krav 31 eller 32, for å generere komprimerte digitale data som er representative for enkeltbilder i videobildeinformasjonen som skal overføres over en standard telefonlinje,

15 k a r a k t e r i s e r t v e d at anordningen for lagring omfatter en enkeltbildegriperanordning for å motta videodata og for å lagre enkeltbilledataene som omfatter billedata som er representative for bildeinformasjonen i ett enkeltbilde av videodata, idet apparatet dessuten omfatter en kontrollanordning koblet til enkeltbildegriperanordningen og telefonlinjen for å levere mengden av identifikatorer for de valgte rekkeviddeblokker til telefonlinjen
20 som komprimerte videodata som representerer enkeltbilledataene.

34. Apparat for å danne videodata for visning på en video-

utgangsinnetning fra en sekvens av forekommende transformasjoner frembragt av apparatet i henhold til krav 31 og mottatt over en standard
25 telefonlinje og hver med et sett av identifikatorer svarende til komprimerte digitale data som er representative for enkeltbildene,

k a r a k t e r i s e r t v e d at apparatet omfatter en datainngangskrets for å motta og akkumulere de komprimerte digitale data,

30 en videodekoder koblet til datainngangskretsen, idet videodekoderen omfatter et minne med en rekke buffere som hver har en rekke adresserbare områder spesifisert av en bufferadresse, og hver av identifikatorene svarer til et adresserbart område i bufferen og innbefatter en bufferadresse og en prosedyrespesifikasjon, og en anordning for å lagre et forhåndsbestemt mønster av digitale data i en av bufferne betegnet som en kildebuffer, å
35 bestemme for hver av identifikatorene et mønster av data svarende til hver identifikator ved å benytte prosedyrene spesifisert i denne identifikator på partiet av kildebufferen angitt av bufferadressen i denne identifikator, å lagre

mønstrene av data i det adresserbare område av en annen av bufferne, betegnet som en destinasjonsbuffer, og angitt i den tilsvarende identifikator, og å avgi innholdet i destinasjonsbufferen som dekomprimerte, digitale data representative for et enkeltbilde av bildeinformasjonen,

5 en videoutgangskrets koblet til videodekoderen for å omdanne digitale data til signaler for visning av et bilde på en videodisplayinnretning,

 og en kontrollanordning koblet til datainngangskretsen og videodekoderen for å levere de akkumulerte digitale data til videodekoderen og for å kommandere videodekoderen til å danne og levere de dekomprimerte,
10 digitale data til videoutgangskretsen.

FIG. 1(a)

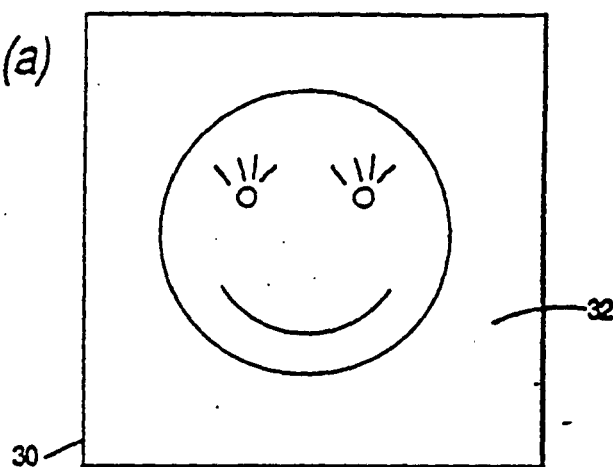


FIG. 1(b)

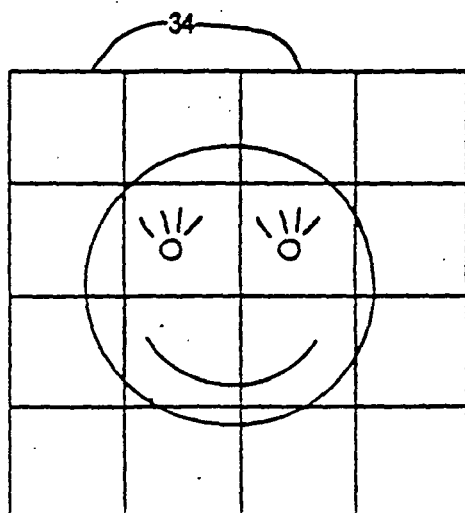


FIG. 1(c)

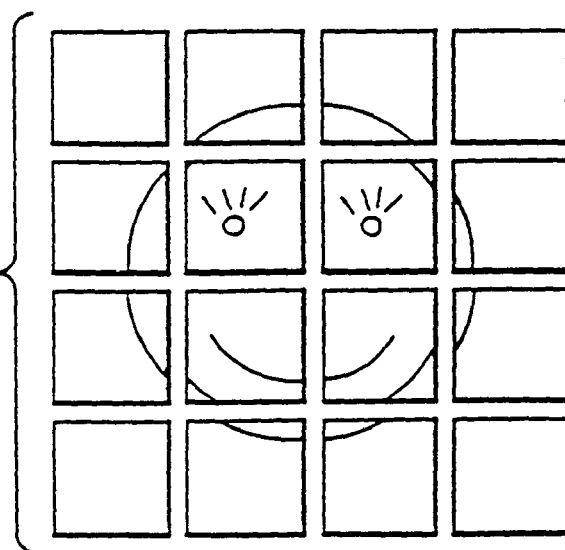


FIG. 2(a)

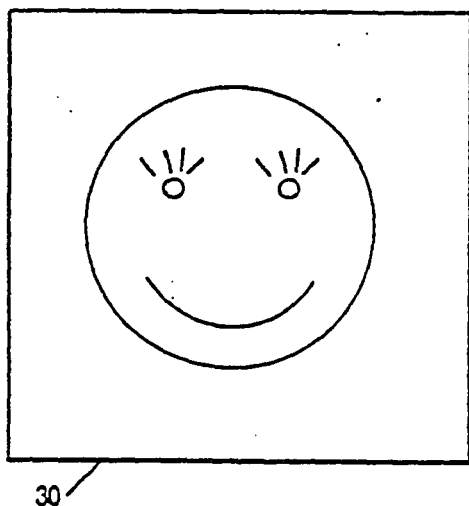


FIG. 2(b)

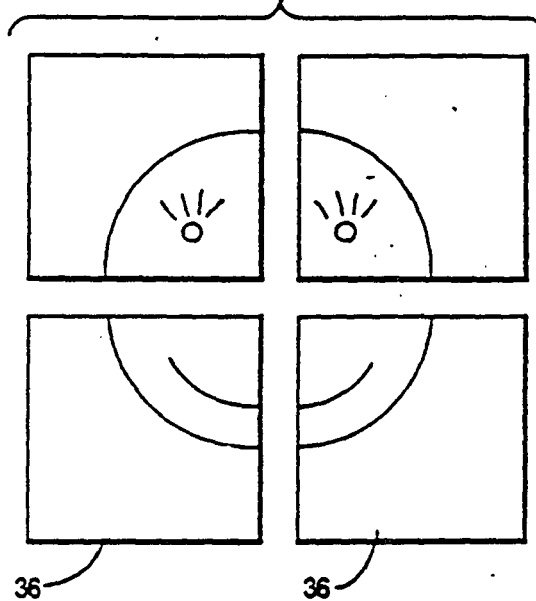


FIG. 2(c)

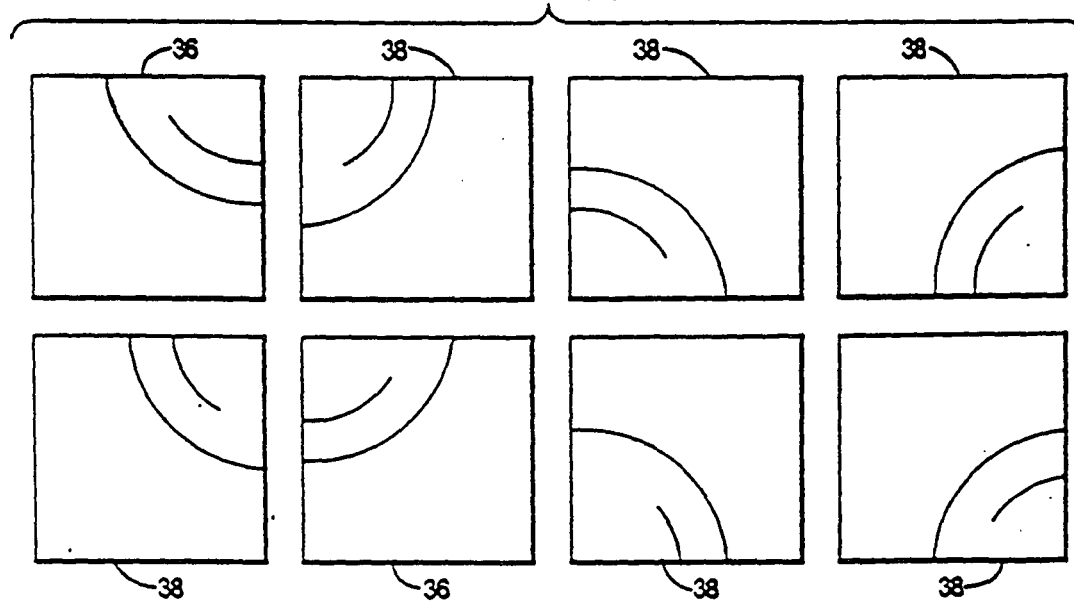


FIG. 3(a)

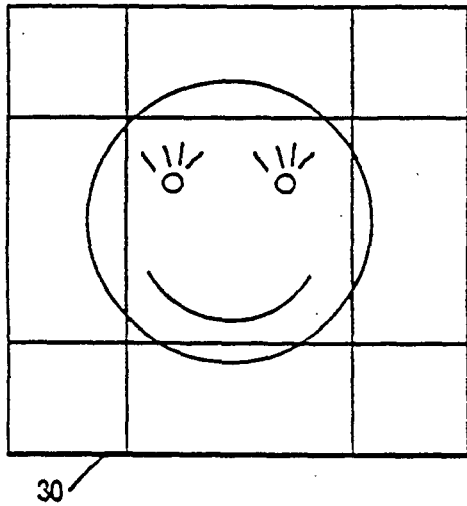


FIG. 3(b)

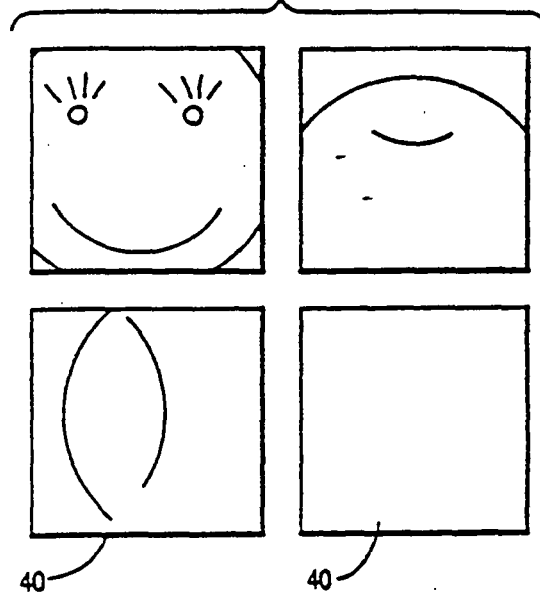


FIG. 4(a)

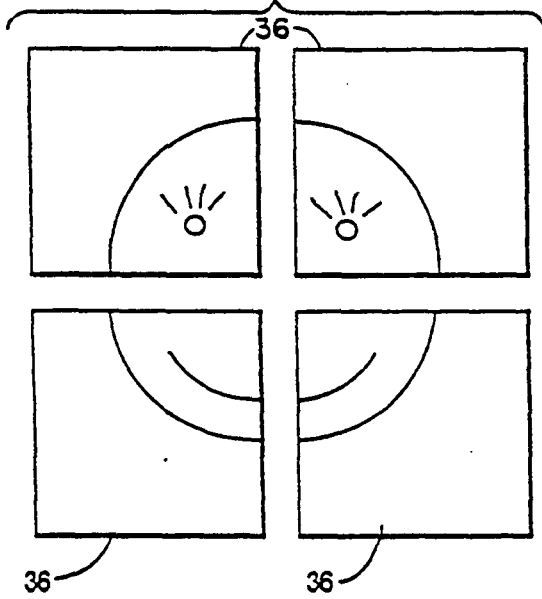


FIG. 4(b)

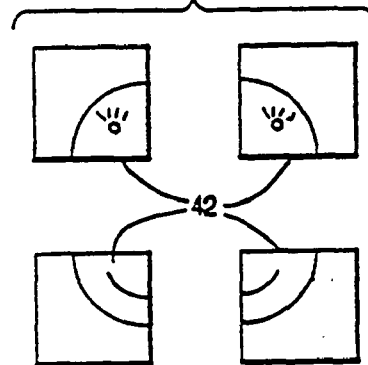


FIG. 5

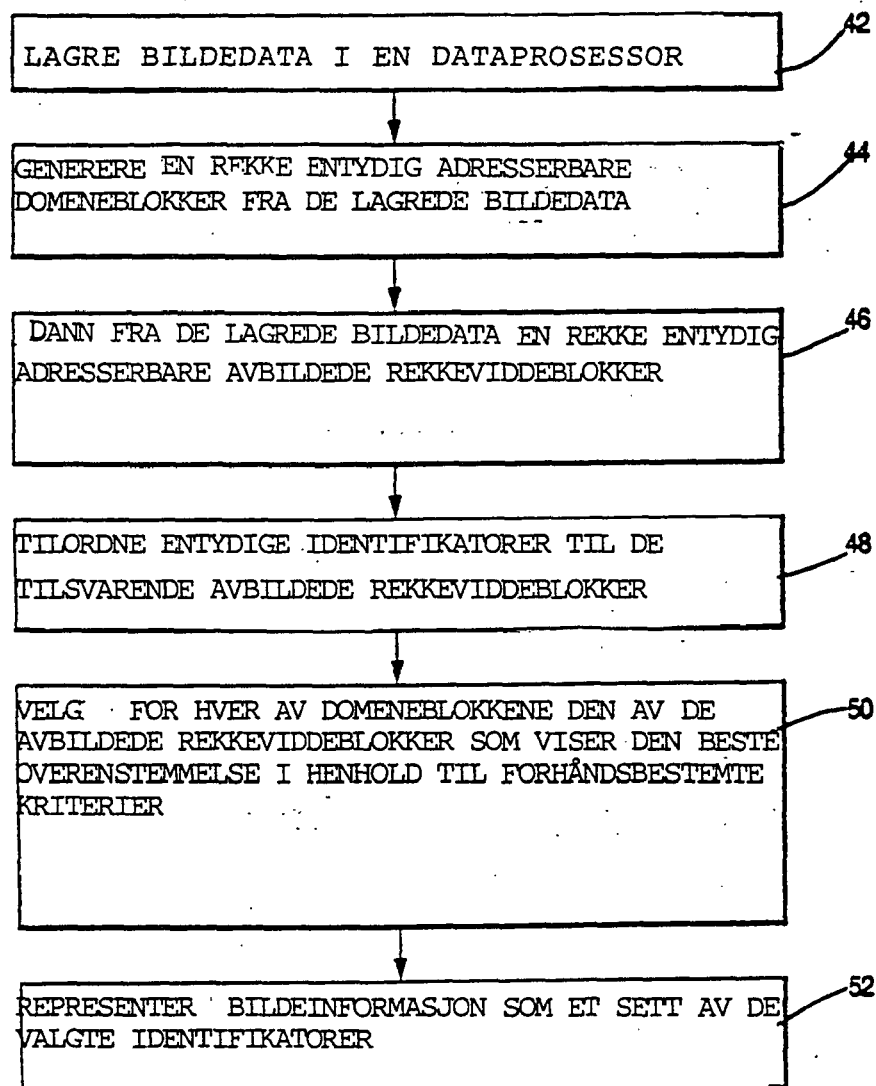
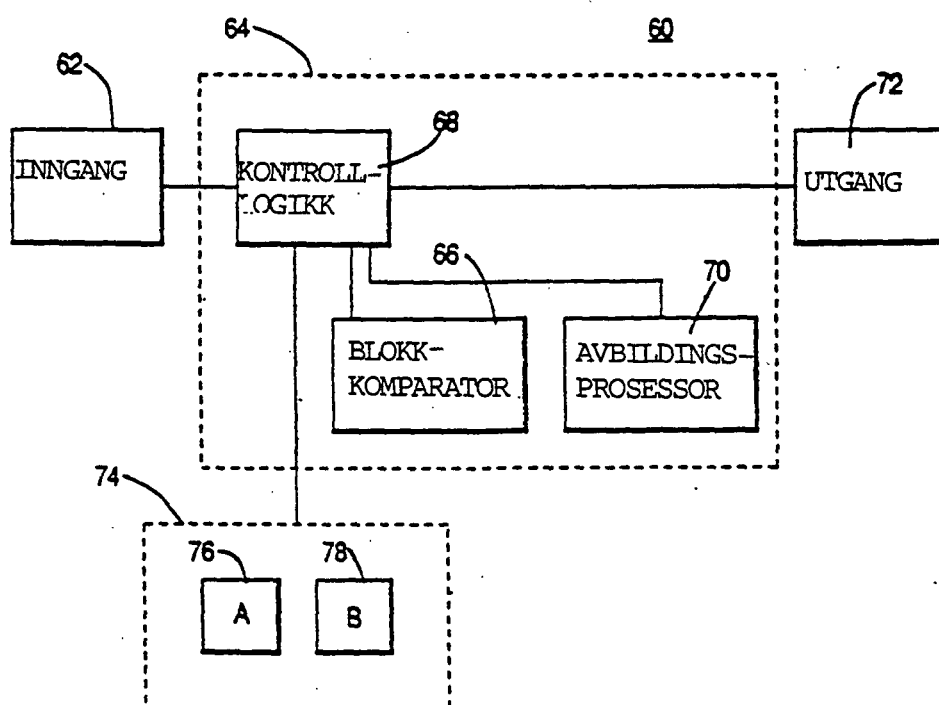


FIG. 6



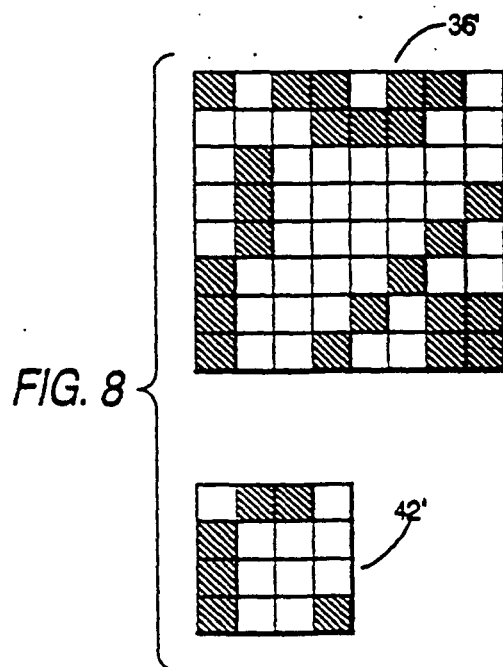
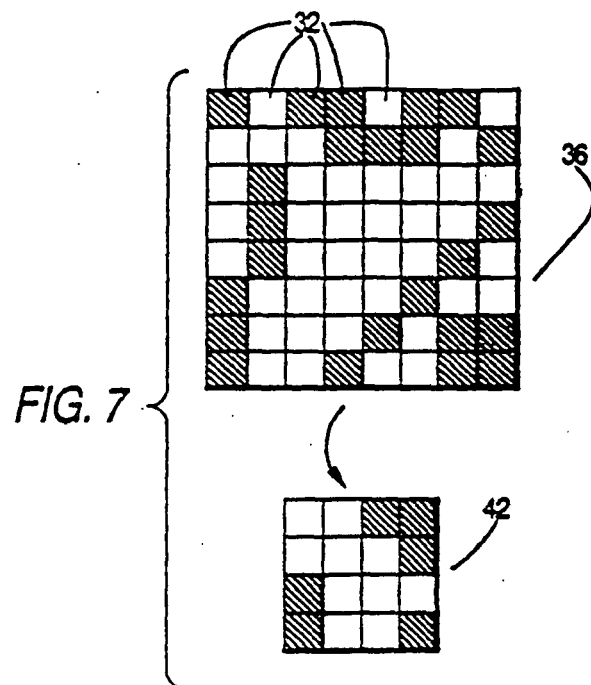


FIG. 9

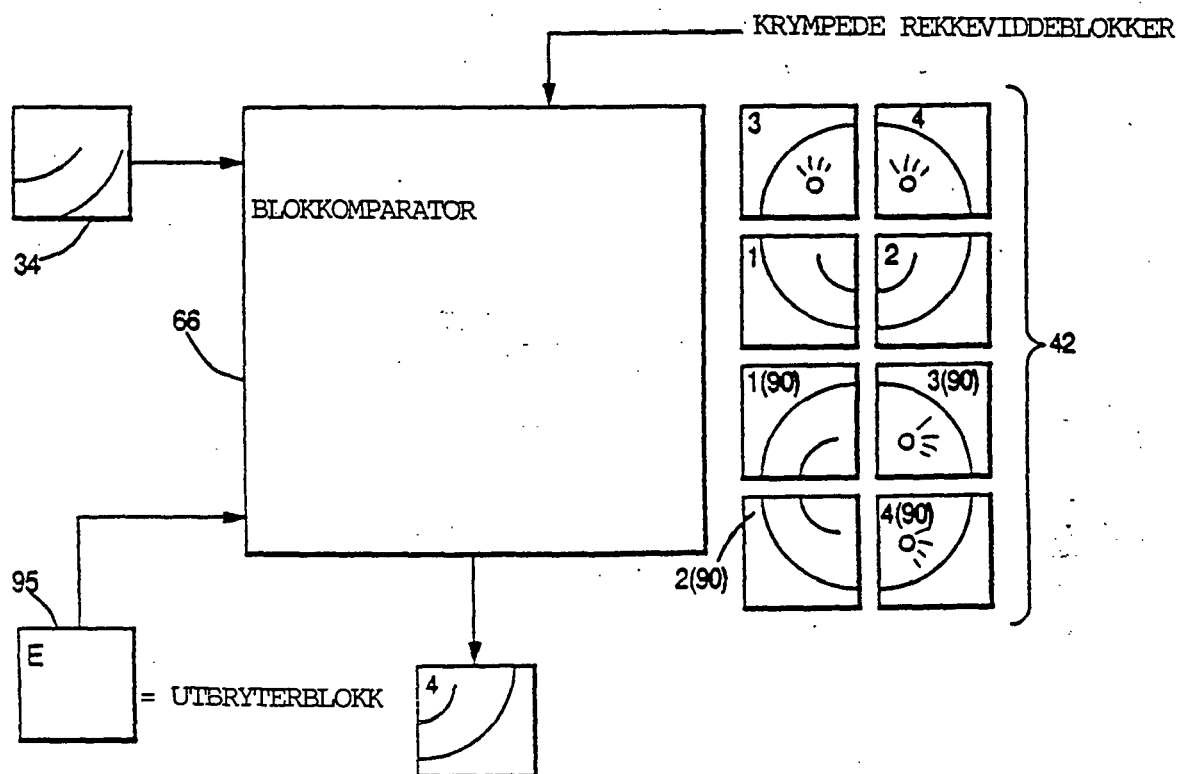
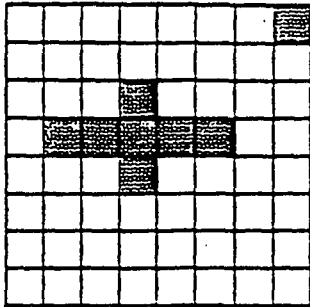


FIG. 10(a)

SVART/HVITT-BLOKK f



BINÆRBLOKK f

0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	1	1	1	1	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

AVSTANDSTABELL FOR f

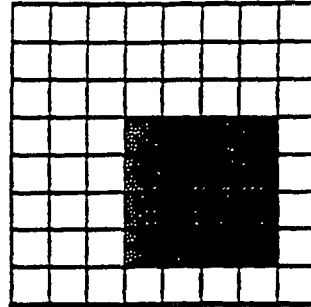
4	3	3	2	3	2	1	0
3	2	2	1	2	2	2	1
2	1	1	0	1	1	2	2
1	0	0	0	0	0	1	2
2	1	1	0	1	1	2	3
3	2	2	1	2	2	3	4
4	3	3	2	3	3	4	5
5	4	4	3	4	4	5	6

AVSTANDSTABELL FOR f
SNITTET BINÆRBLOKK FOR g

			0	0	0	1	
			0	1	1	2	
			1	2	2	3	
			2	3	3	4	

FIG. 10(b)

SVART/HVITT-BLOKK g



BINÆRBLOKK g

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0

AVSTANDSTABELL FOR g

5	4	3	3	3	3	3	4
4	3	2	2	2	2	2	3
3	2	1	1	1	1	1	2
3	2	1	0	0	0	0	1
3	2	1	0	0	0	0	1
3	2	1	0	0	0	0	1
3	2	1	0	0	0	0	1
4	3	2	1	1	1	1	2

AVSTANDSTABELL FOR g
SNITTET BINÆRBLOKK FOR f

							4
			1				
	2	1	0	0	0		
		0					

FIG. 11

The diagram shows a 4x4 grid of blocks. Each block contains the text 'ADRESSE TIL KRYMPET REKKEVIDDE-BLOKK' followed by a specific identifier in brackets. The first row has identifiers [3], [5(90)], [E], and an empty space. The second, third, and fourth rows have empty spaces in all four columns. Above the first two columns, there is a bracket labeled '99'. Above the last two columns, there is another bracket labeled '99'. To the right of the grid, there is a curved arrow pointing downwards, labeled '98'.

ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK [3]	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK [5(90)]	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK [E]	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK
ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK
ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK
ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK	ADRESSE TIL KRYMPET REKKEVIDDE- BLOKK

FIG. 12

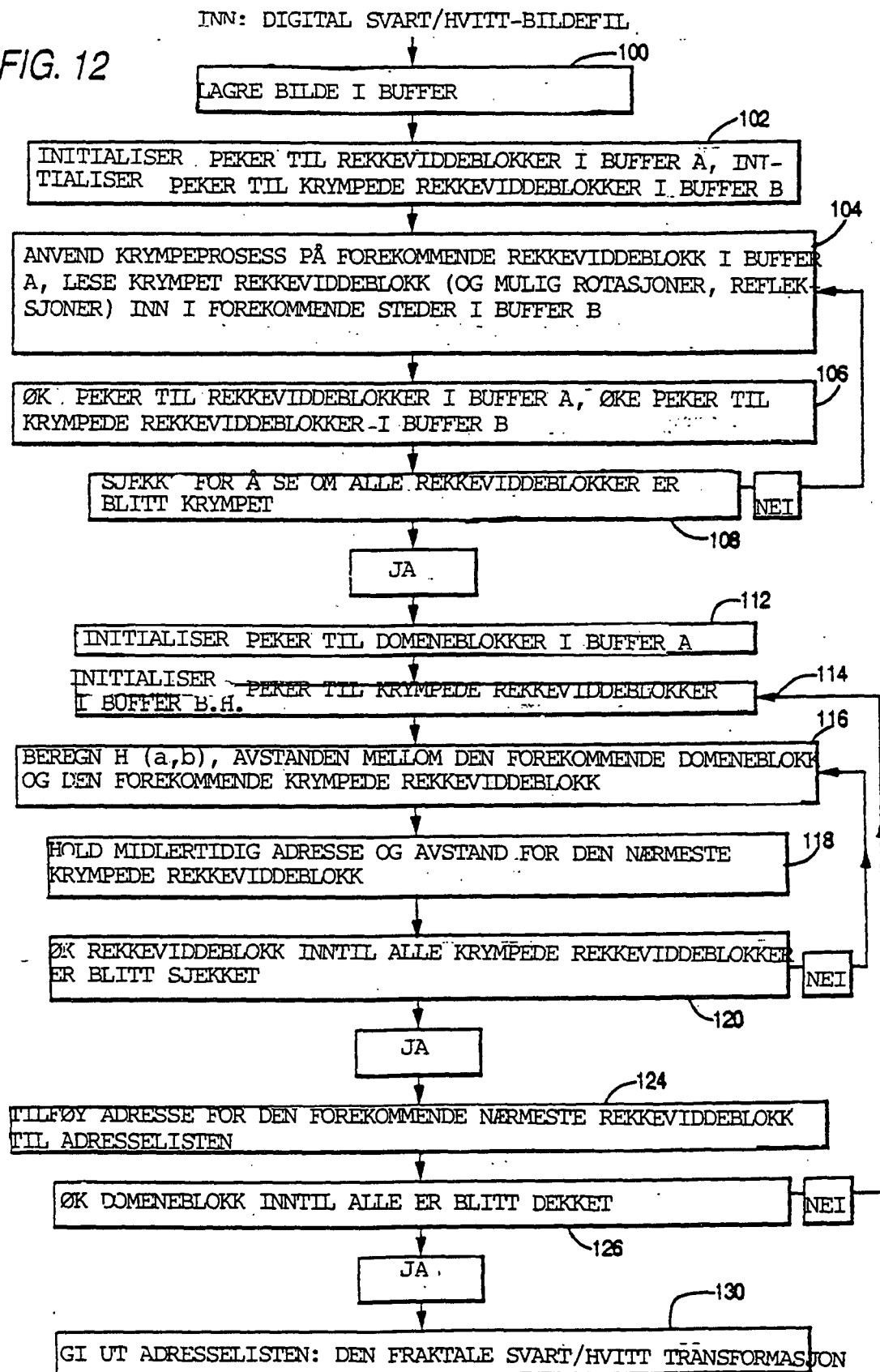


FIG. 13

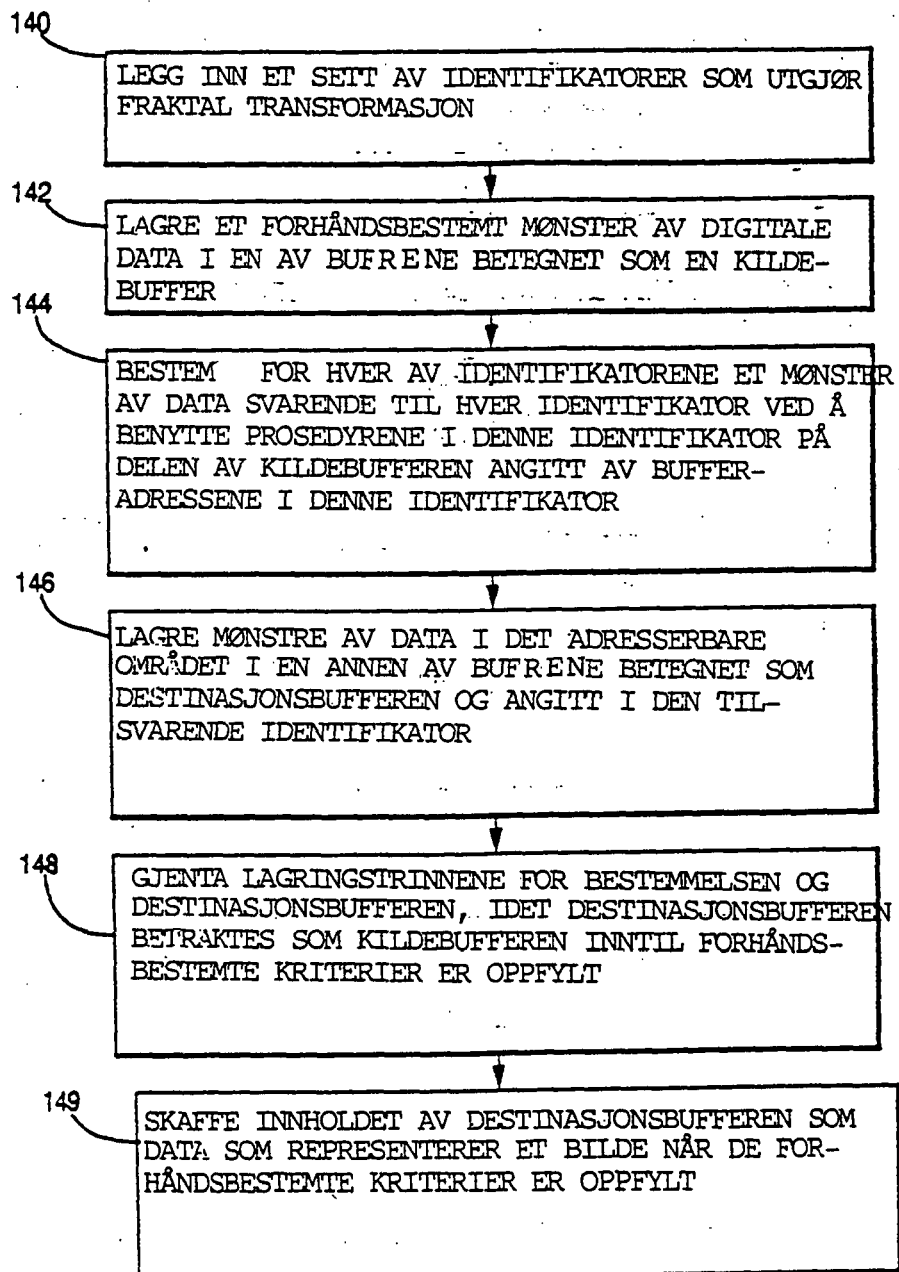


FIG. 14

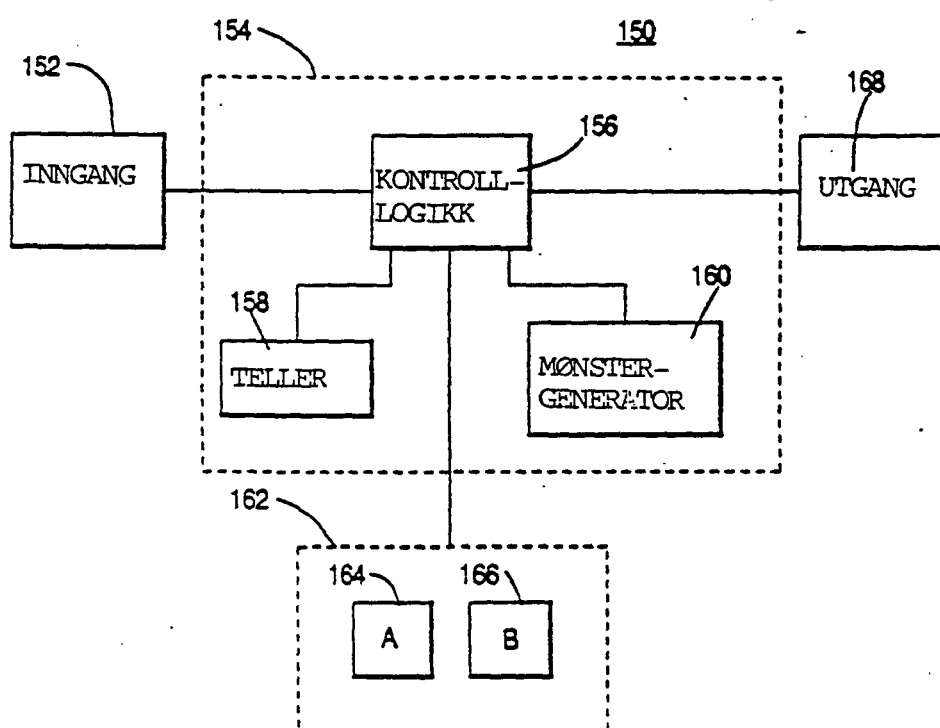


FIG. 15

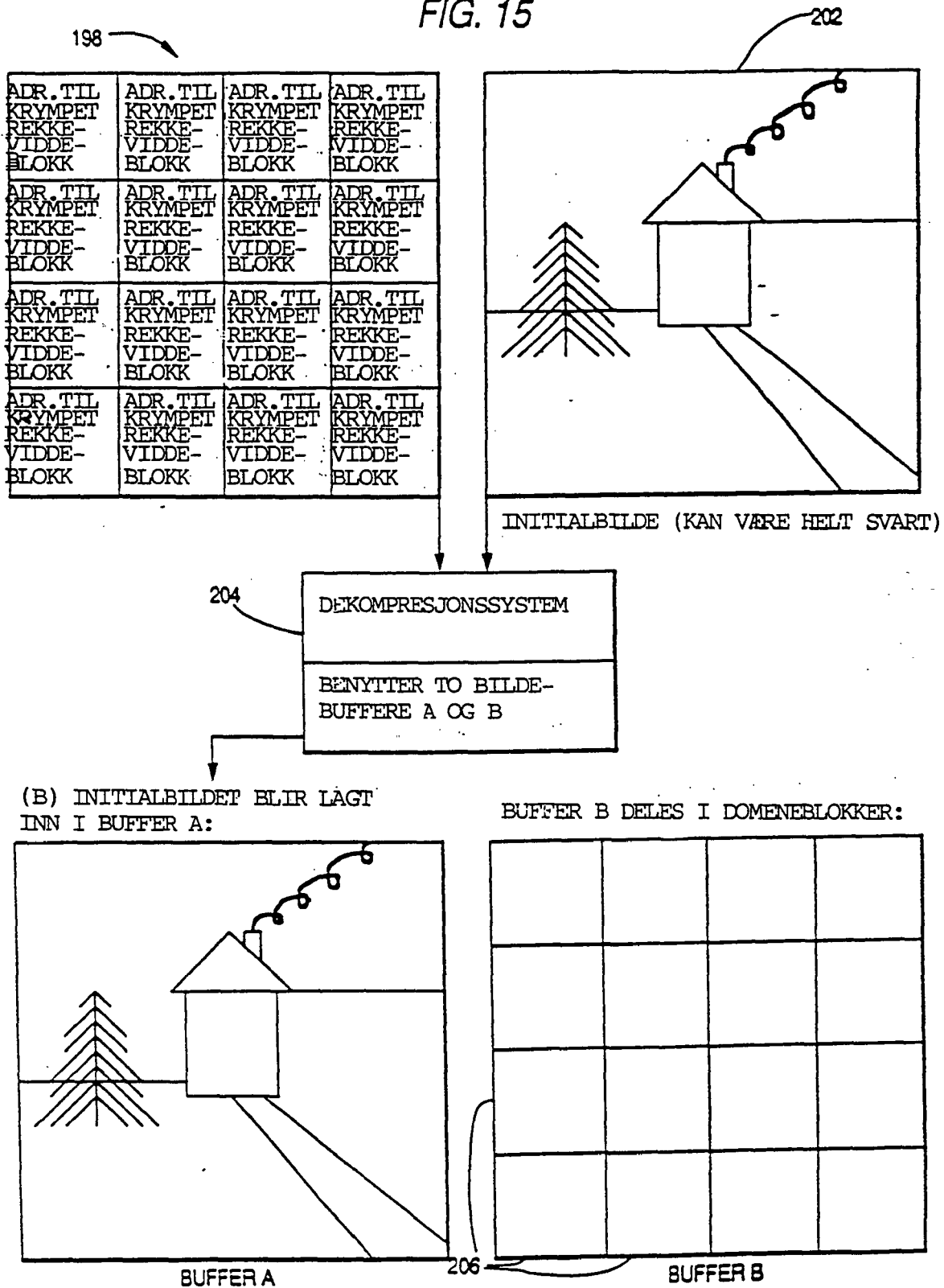


FIG. 16

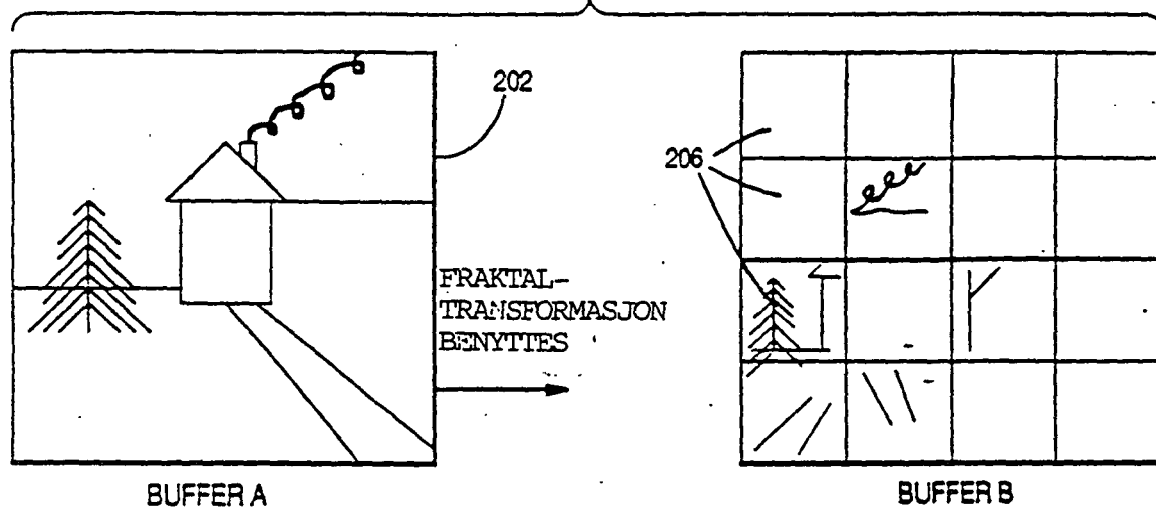


FIG. 17

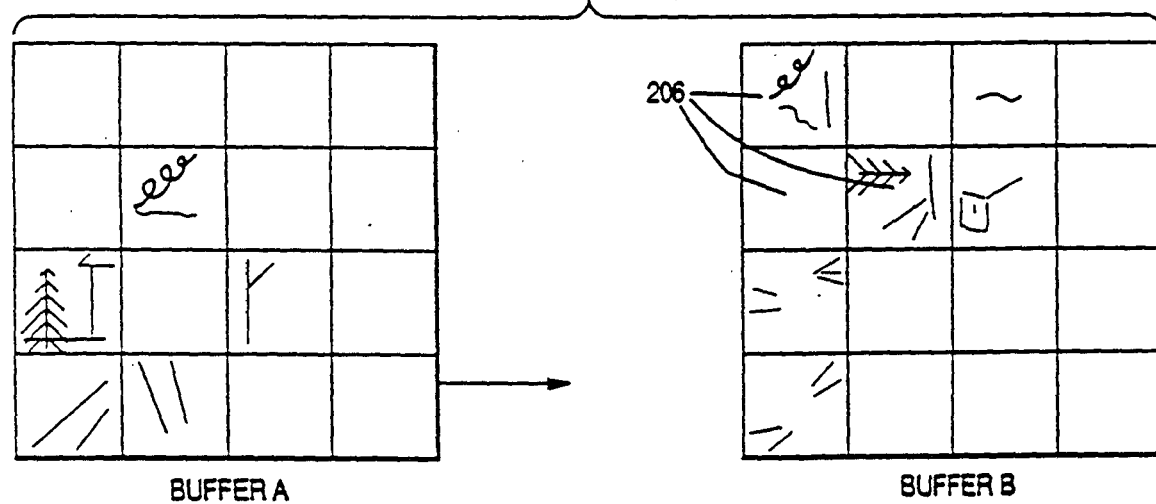


FIG. 18

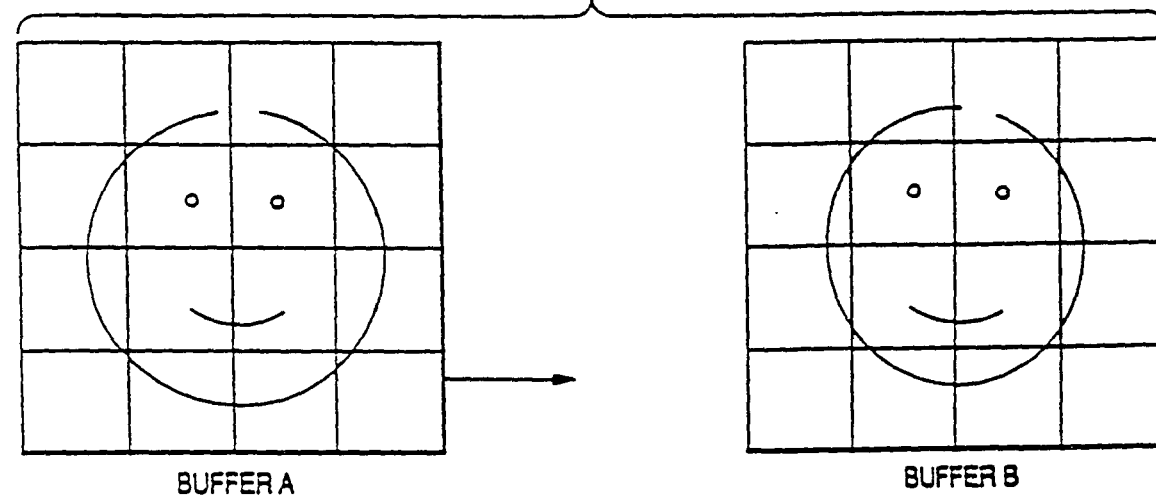


FIG. 19

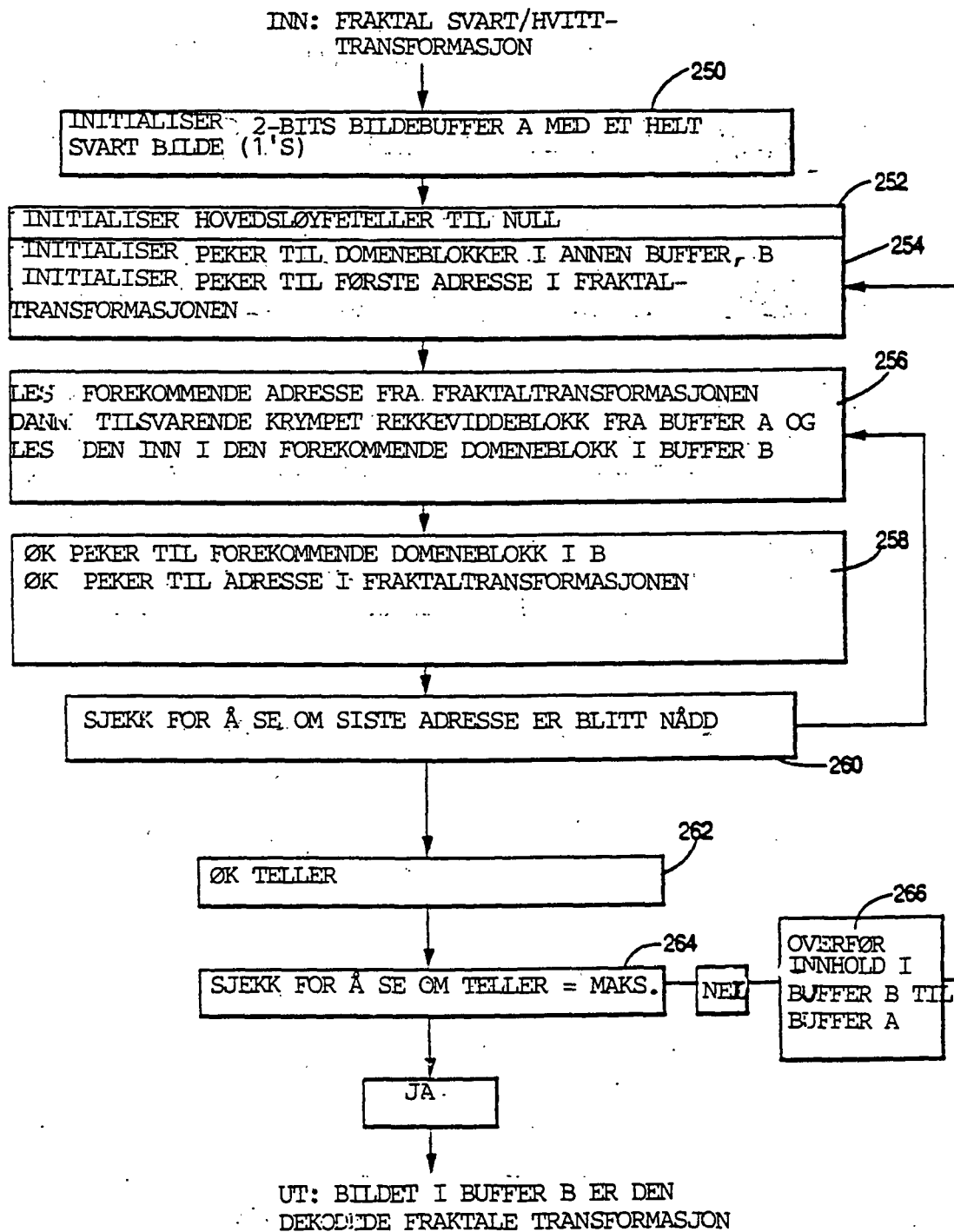


FIG. 20

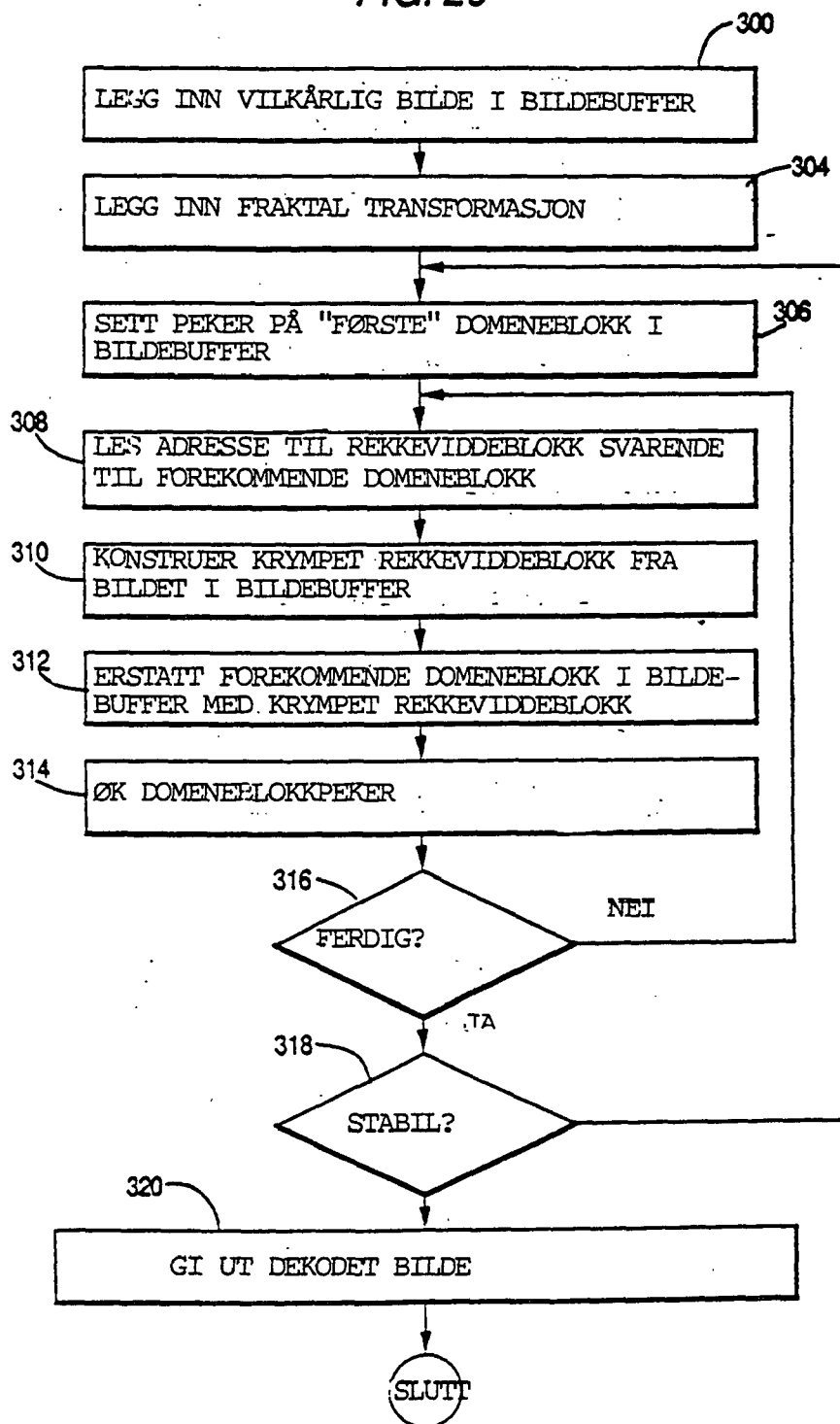


FIG. 21(a)

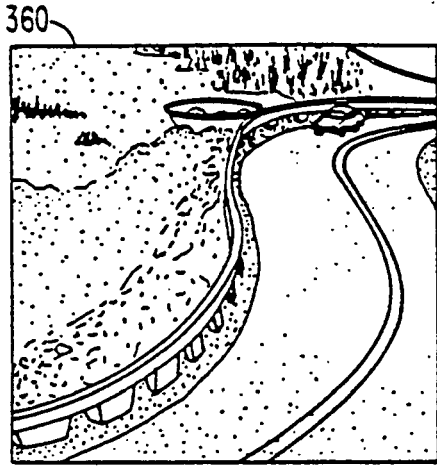


FIG. 21(b)

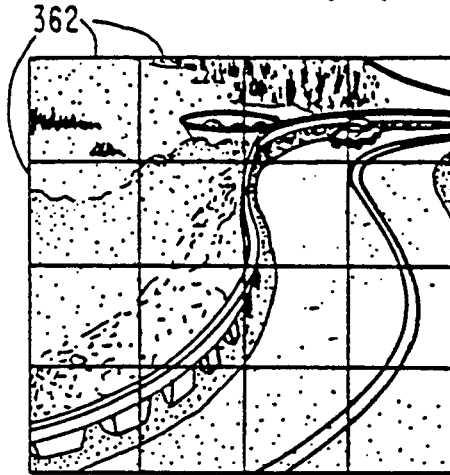
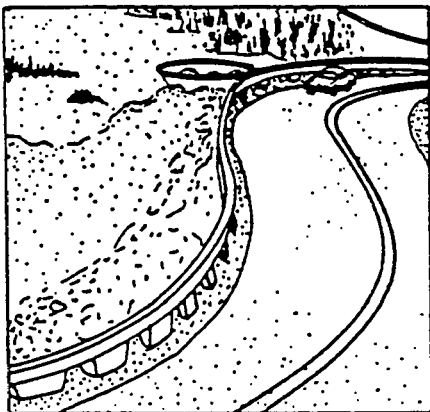
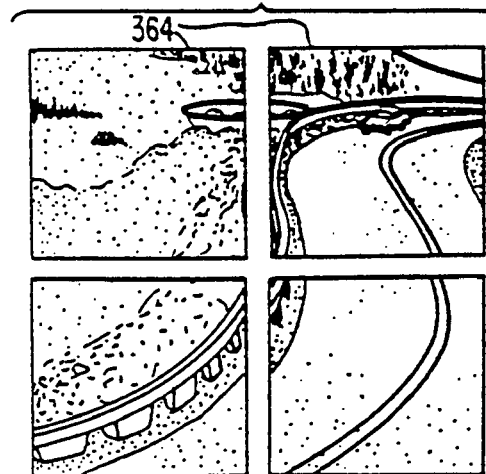


FIG. 22(a)



360

FIG. 22(b)



364

FIG. 23(a)

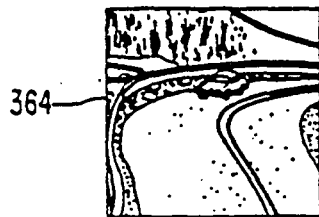


FIG. 23(b)



FIG. 24

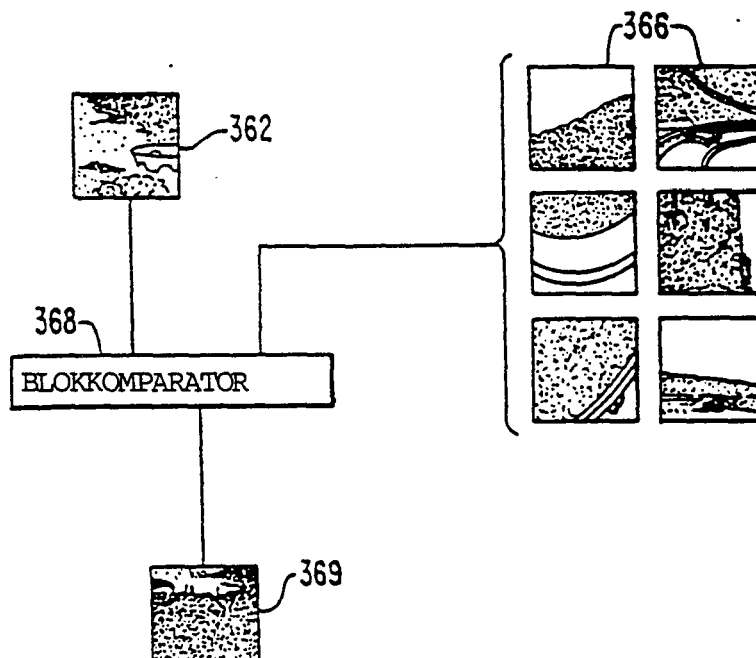


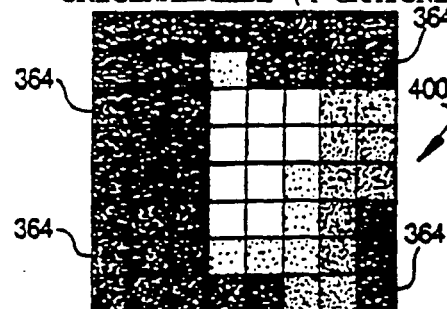
FIG. 25

ORIGINALBILDE (BINÆRT)

2	2	2	2	2	2	2	3
2	2	2	0	2	2	2	2
2	2	2	0	0	1	2	2
3	2	2	0	0	1	2	2
3	2	2	0	0	1	2	2
3	3	2	0	0	1	2	2
3	3	2	1	1	1	2	2
3	3	3	3	3	2	2	2

402

ORIGINALBILDE (4 GRÅTONER)



KRYMPEDE

REKKEVIDDEBLOKKER

3	1
3	2

#1

XTRANS=0
YTRANS=0

0	2
2	2

#2

XTRANS=4
YTRANS=0

2	1
2	1

#3

XTRANS=0
YTRANS=4

2	2
0	2

#4

XTRANS=4
YTRANS=4

404

DOMENEBLOKKER

3	3
3	3

#1

2	1
3	3

#2

3	2
3	3

#5

2	0
2	0

#6

2	2
3	2

#9

2	0
2	0

#10

2	2
2	2

#13

2	2
2	0

#14

1	1
3	2

#3

0	1
0	1

#7

0	1
0	1

#11

2	2
2	2

#15

2	3
2	3

#4

2	2
2	3

#8

2	2
2	2

#12

2	3
2	2

#16

406

FRAKTAL GRÅSKALATTRANSFORMASJON

DOMENEBLOKK ~~≠~~REKKEVIDDEBLOKK ~~≠~~

P

Q (AVSTAND)

1	1	0.5	1	(10)
2	1	0.5	1	(2)
3	1	0.5	0	(3)
4	1	0.5	1	(6)
5	1	0.5	1	(6)
6	1	0.5	1	(1)
7	2	0.5	0	(1)
8	1	0.5	0	(6)
9	1	0.5	0	(5)
10	1	0.5	0	(1)
11	2	0.5	0	(1)
12	3	0.5	1	(2)
13	3	0.5	1	(2)
14	1	0.5	0	(3)
15	3	0.5	1	(3)
16	2	0.5	1	(2)

408

FIG. 26

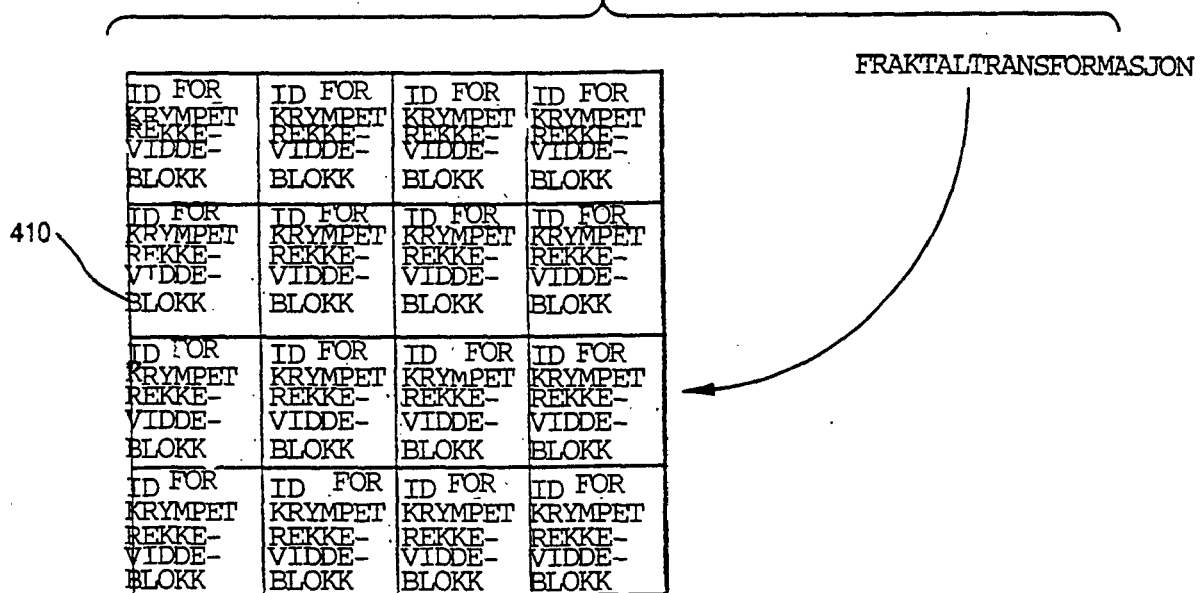


FIG. 27

INN: DIGITAL GRÅSKALABILDEFIL

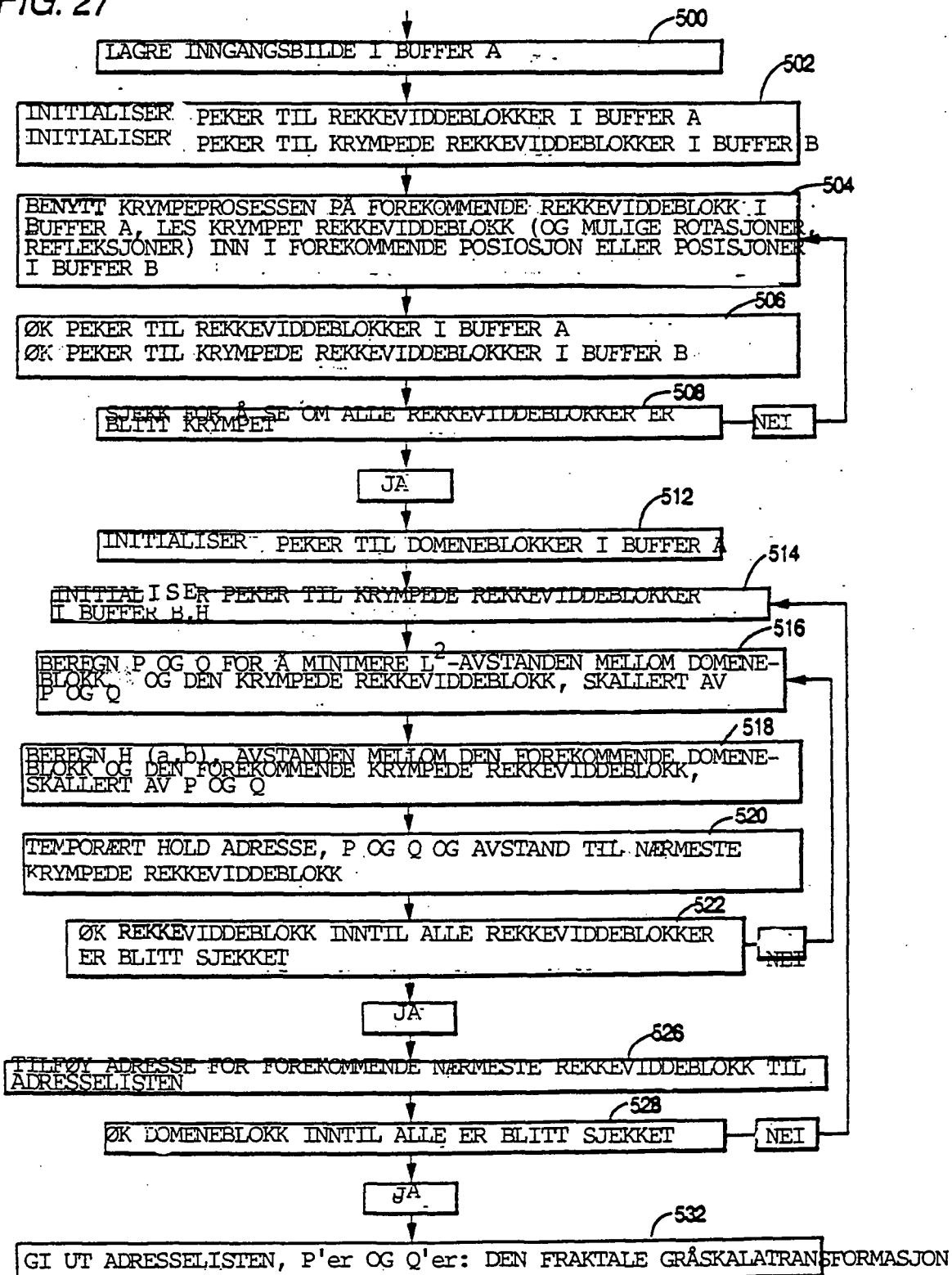


FIG. 28

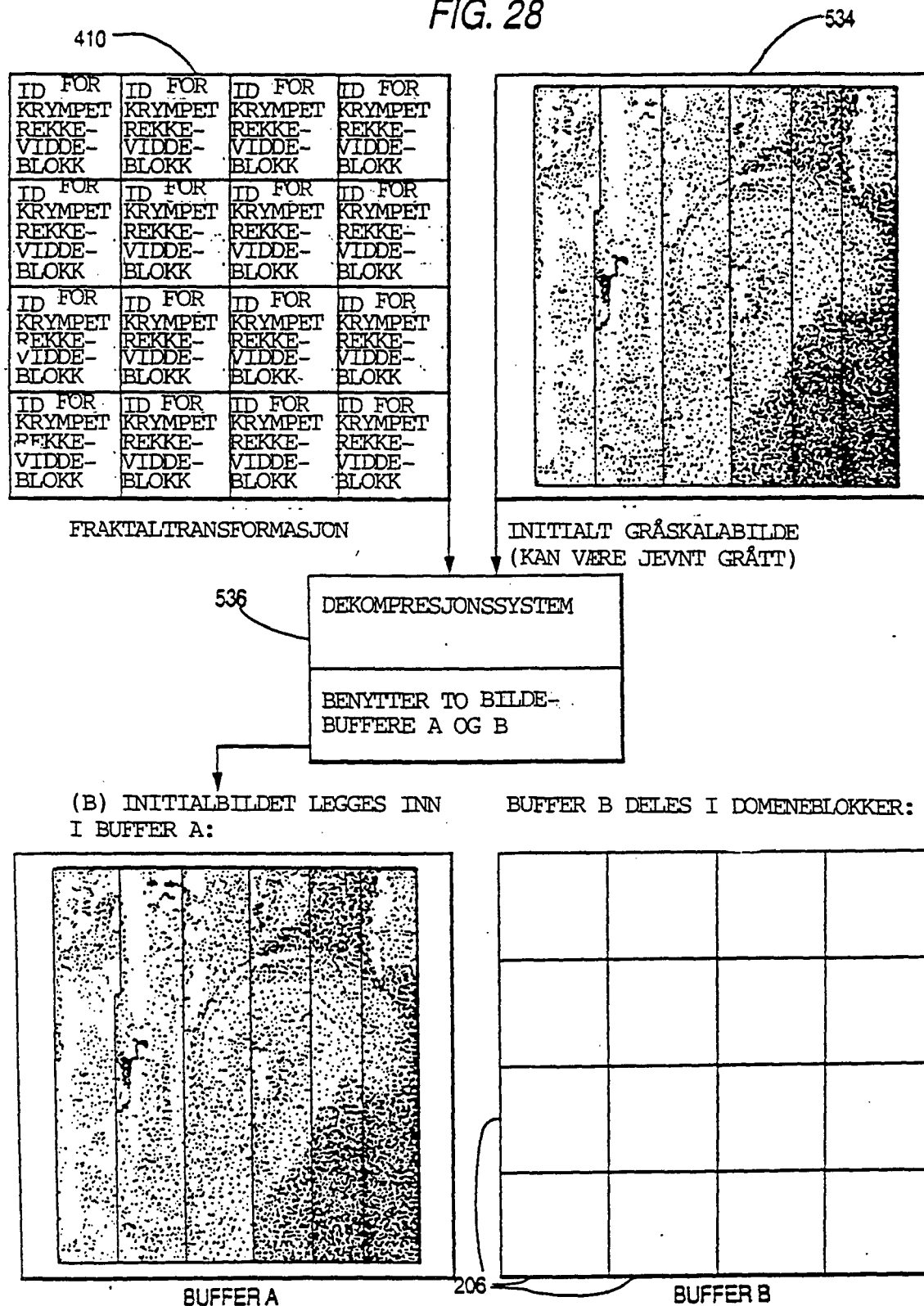


FIG. 29

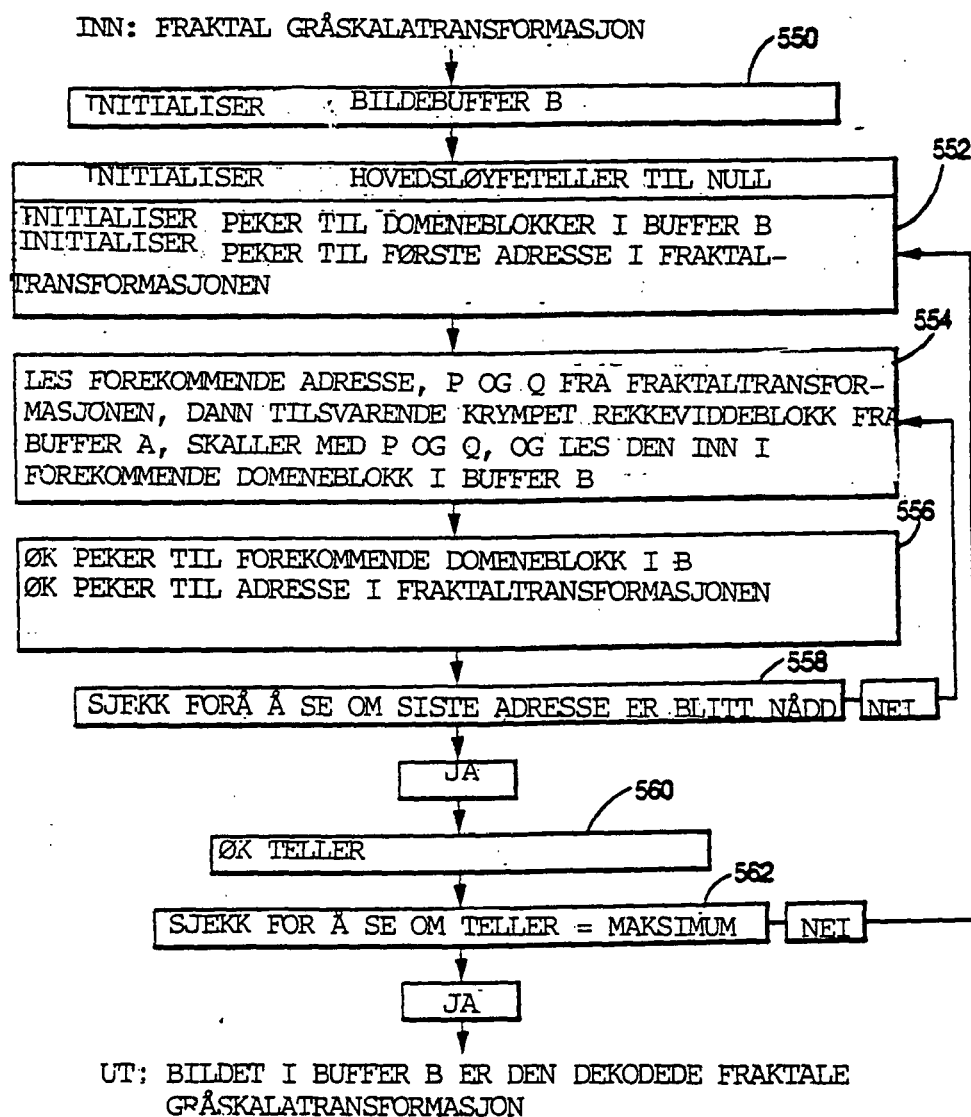


FIG. 30

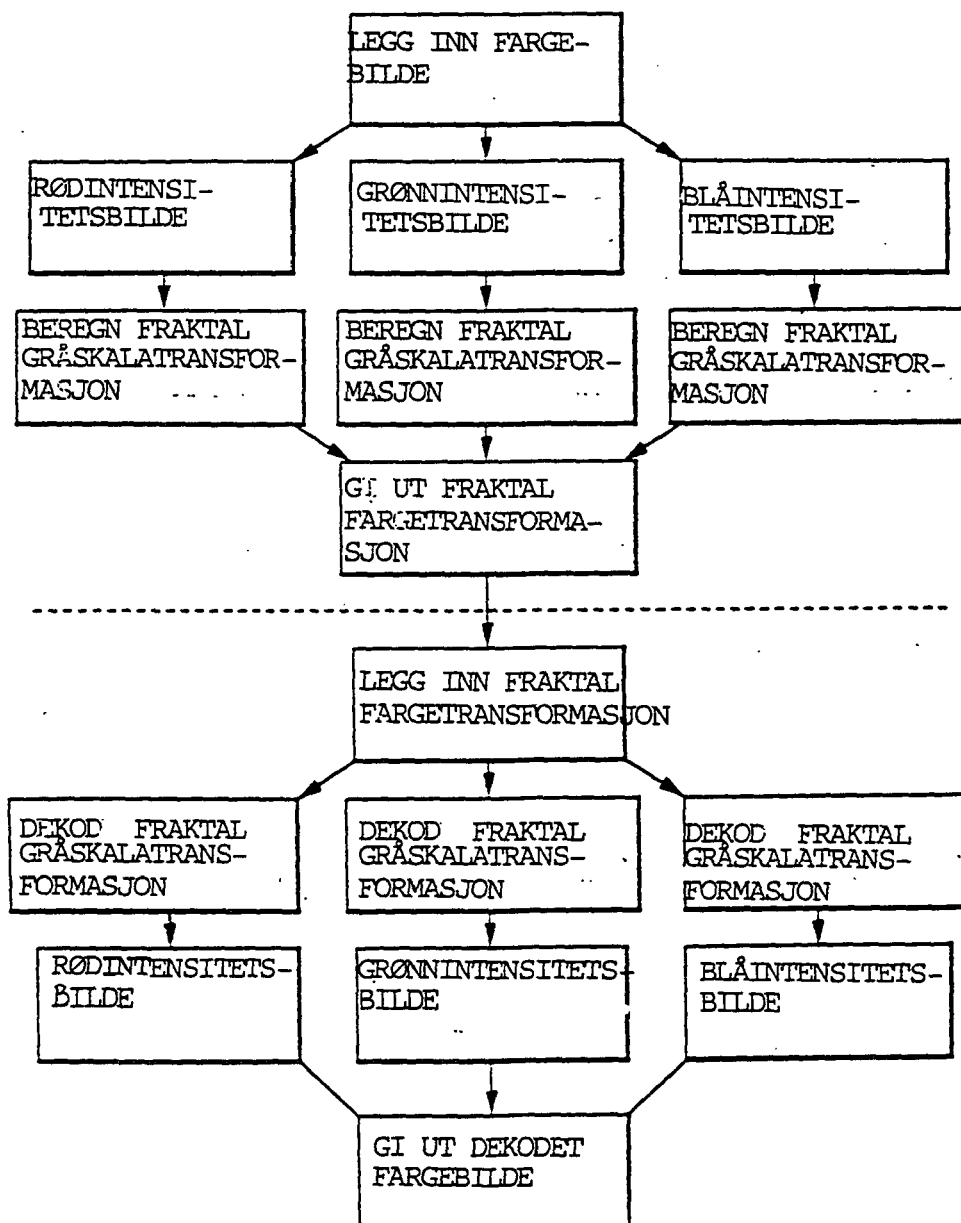


FIG. 31

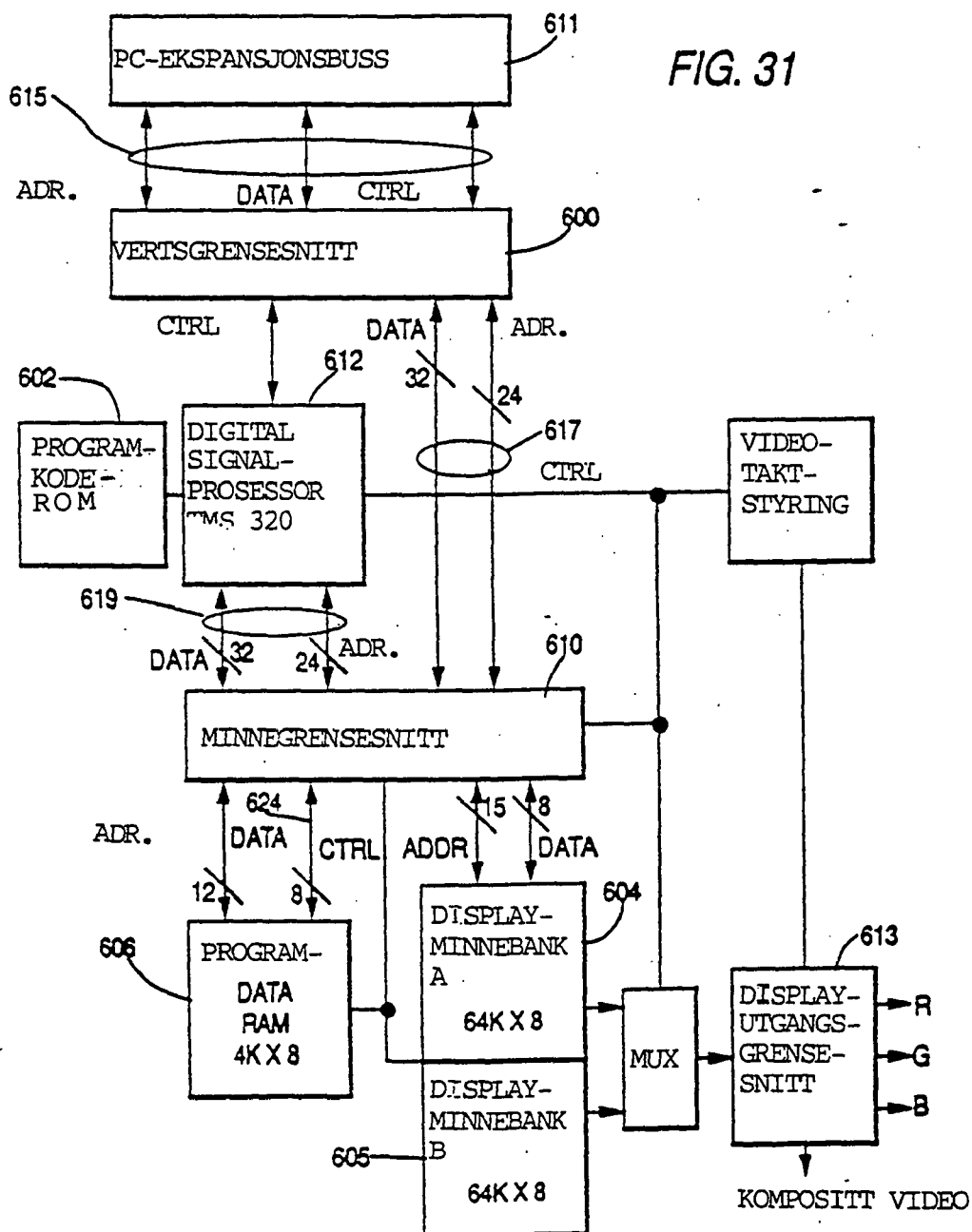


FIG. 32

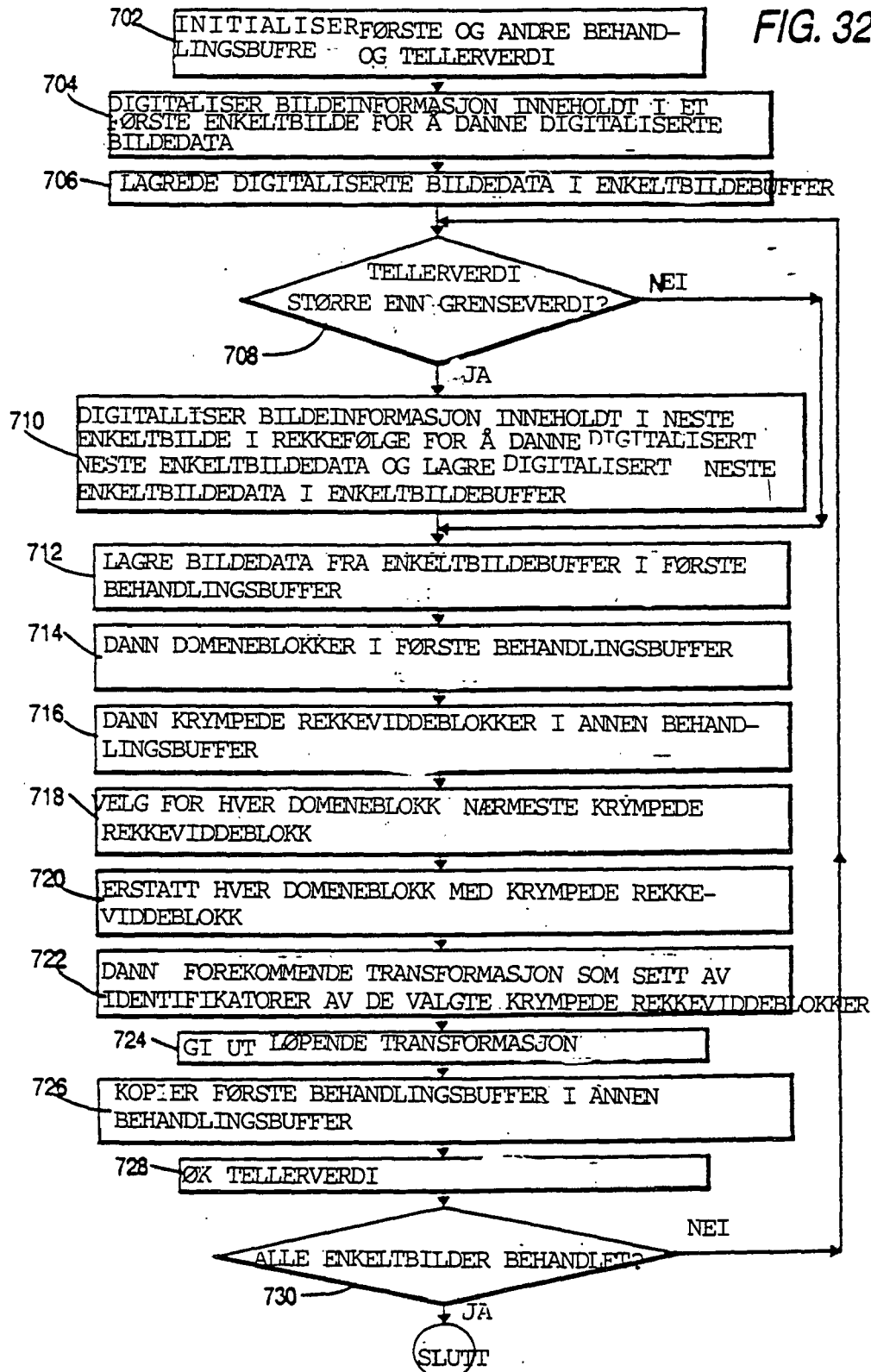


FIG. 33

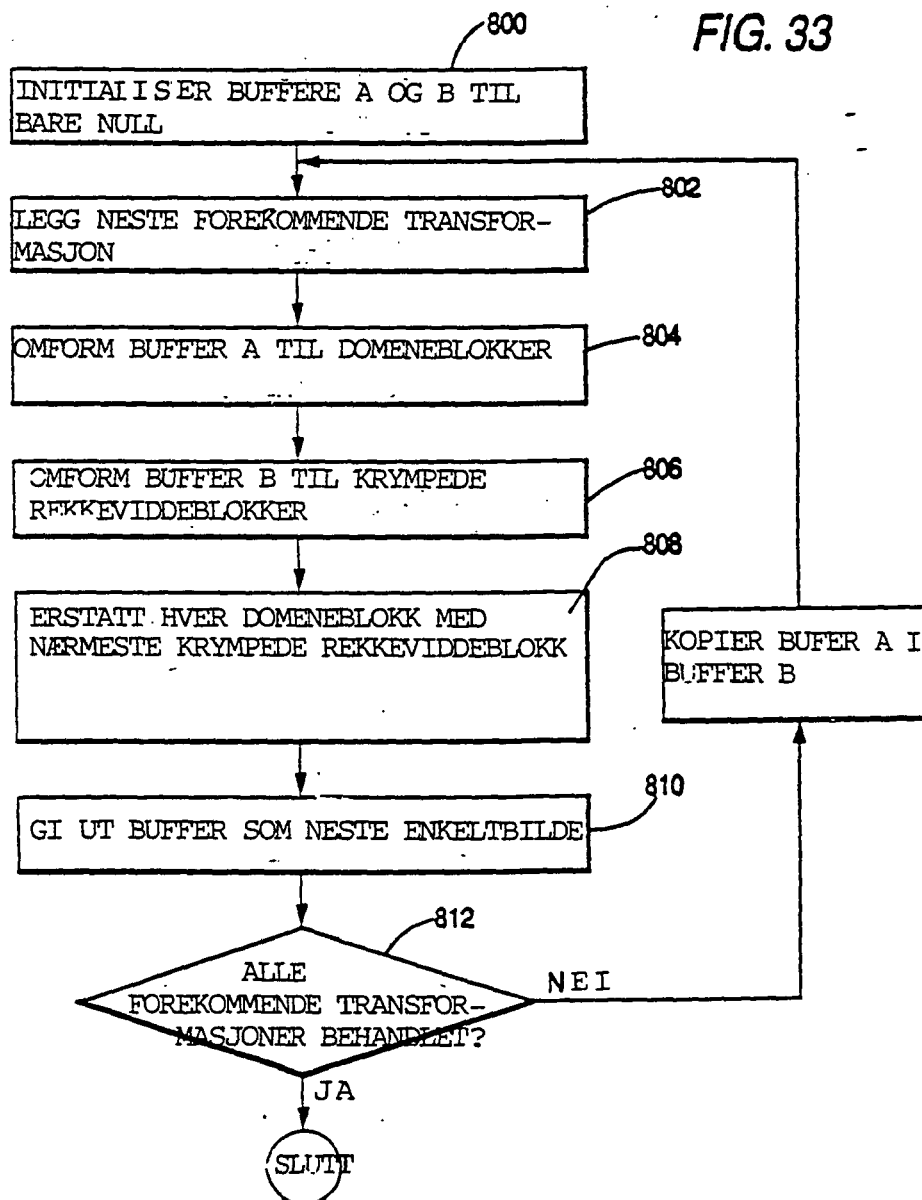


FIG. 34

