



(12) PATENT

(19) NO

(11) 338043

(13) B1

NORGE

(51) Int Cl.

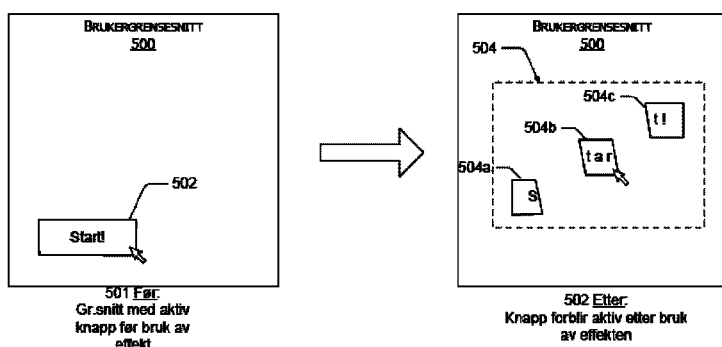
G06F 3/048 (2013.01)

## Patentstyret

(21)	Søknadsnr	20080709	(86)	Int.inng.dag og søknadsnr	2006.09.05 PCT/US2006/34613
(22)	Inng.dag	2008.02.08	(85)	Videreføringsdag	2008.02.08
(24)	Løpedag	2006.09.05	(30)	Prioritet	2005.09.13, US, 60/716,735 2006.07.14, US, 11/457,711
(41)	Alm.tilgj	2008.04.14			
(45)	Meddelt	2016.07.25			
(73)	Innehaver	Microsoft Technology Licensing, LLC, One Microsoft Way, US-WA98052 REDMOND, USA			
(72)	Oppfinner	Adam M Smith, c/o Microsoft Corp, One Microsoft Way, US-WA98052 REDMOND, USA Robert A Wlodarczyk, c/o Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA Biliana K Kaneva, c/o Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA Eduardo M Maia, c/o Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA Patrick J Sweeney, c/o Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA Rahul V Patil, c/o Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA Sriram Subramanian, c/o Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA			
(74)	Fullmektig	Bryn Aarflot AS, Postboks 449 Sentrum, 0104 OSLO, Norge			

(54)	Benevnelse	<b>Utvidbare synlige effekter for aktivt innhold i brukergrensesnitt</b>
(56)	Anførte publikasjoner	US 5490246 A US 6522331 B1 EP 977157 A2 EP 977146
(57)	Sammendrag	

Fremgangsmåter og systemer for å anvende visuelle effekter på aktivt innhold, så som knapper, kombinasjonsbokser, video, redigeringsfelter, etc., som bevarer interaktiviteten til det aktive innholdet. Foreliggende oppfinnelse tilveiebringer også en mekanisme som lar utviklere bygge opp nye visuelle effekter og anvende dem på aktivt innhold.



## BAKGRUNN

[0001] Noen ganger ønsker applikasjonsutviklere å implementere brukergrensesnitt som inneholder unike elementer som skiller brukergrensesnittet fra andre brukergrensesnitt, og/eller å forbedre en brukers opplevelse når brukeren jobber med applikasjonen via brukergrensesnittet. Én måte en utvikler kan ønske å individuelt tilpasse et brukergrensesnitt på er ved å skape visuelle effekter som opptrer for å signalisere en endring i et brukergrensesnitt eller for å modifisere et element i et brukergrensesnitt. Dagens plattformer støtter imidlertid ikke bruk av en visuell effekt på innhold som anvendes som del av et brukergrensesnitt.

[0002] EP 977146 beskriver en digital videobehandlingsanordningen som omfatter: en flerhet av visningsprosessorer arrangert i en operasjonssekvens, som hver kan opereres for å vise et utgangresultat vedrørende et bilde fra et videosignal fra inngangsdata knyttet til dette og / eller andre bilder mottatt fra en forutgående visningsprosessor i den operasjonelle sekvensen; hver visningsprosessor kan opereres for å oppdage og kommunisere med andre visningsprosessorer om dens viste utgang gjengitt er konstant mellom tilstøtende bilder.

[0003] EP 977157 A2 beskriver et apparat for videospesialeffekter som omfatter et kjerneprogramobjekt; og en flerhet av visningsprosessor plug-in program objekter som responderer på kjerneprogramobjektet, visningsprosessorer er anordnet i rettet asyklisk graf, som hver kan betjenes for å gi et utgangresultat i forbindelse med suksessive bilder fra et videosignal fra inngangsdata som gjelder dette og / eller andre bilder mottatt fra en forutgående visningsprosessor i den rettede asyklisk graf, og som responderer på operasjonsparametere relatert til denne visningsprosessoren; hvor hver operasjonsparameter er definert av et parameter plug-in objekt innrettet for å kommunisere en parameterverdi til visningsprosessoren's plug-in objekt.

[0004] US 6522331 B1 beskriver et figuranimasjonssystem, metode og datamaskinprogramprodukt for å animere et skjelett. Skjelettet har ben som er anordnet i et hierarki. Hvert ben i skjelettet er definert i forhold til det overordnede ben og det øverste benet i hierarkiet er rot-benet. Når et ben i skjelettet er animert, blir alle ben under det animerte benet i hierarkiet også

animert. En animasjon av et ben er spesifisert med en rettet asyklisk graf (DAG) som har en eller flere noder. Nodene er enten datagenereringsnoder (DG noder) eller kombineringsfilter noder. DG noder har null eller én inngang og har en utgang som sender ut animasjons data. DG noder som har null innganger inkluderer faste forhånds-genererte, sporings, revers, speil, og støy noder. DG noder som har en inngang inkluderer begrensning og reflekterende noder. Kombineringsnoder har to innganger for å motta animasjons data og en utgang som sender ut animasjons data. Kombineringsnoder inkluderer legge over hverandre, blande, og transisjons noder. Når et ben animeres, spør benet sin tilhørende DAG, om noen, og gir DAG tiden som har gått siden forrige spørringen. Den øverste noden av DAG spør rekursivt lavere noder inntil DG noder er nådd og animasjonsdataen når toppnoden. Disse animasjonsdataene animerer benet. Ved bruk av DAGer på denne måten vil overgangene mellom ulike animasjoner bli glatte som gjør at repeterende animasjoner vises mer realistisk.

[0005] US 5490246 A beskriver en grafikk editor som genererer en utseende konstruksjonsgraf (ACG) som representerer den dannelsesprosessen av et bilde som en rettet asyklisk graf (DAG). ACG bildetransformasjonsoperasjoner (Transforms), og utganger (synbare bilder) som kan vises, for eksempel i et vindu på en skjerm, kan manipulert for å konstruere og redigere komplekse bilder. Følgelig er det viste ACG et brukergrensesnitt som tillater komplekse grafikkbilder i å bli bygget på en enkel måte, og dermed lett forstått og manipulert (redigert) av en operatør. Nodene i ACG er selektivt sammenkoblet med linker (kanter). Det komplekse bildet representert i ACG kan konstrueres og vises i et visningsvindu på skjermen ved å "kjøre" ACG. Grafikkeditoren "kjører" ACG ved å utføre operasjoner på bildefragmenter som er definert av sammenkoblede noder av ACG. Når ACG er "kjørt", vil hver utgang eller visningsnode vise sitt bilde i et visningsvindu som opprettes automatisk.

30

## OPPSUMMERING

[0006] Det følgende presenterer en forenklet oppsummering av oppfinnelsen for å gi leseren en grunnleggende forståelse. Denne

oppsummeringen er ikke en utfyllende oversikt over oppfinnelsen, og den er ikke ment å identifisere nøkkelementer/avgjørende elementer i oppfinnelsen eller å avgrense oppfinnelsens ramme. Dens eneste formål er å presentere i en forenklet form noen idéer som er beskrevet her som en innledning til den mer detaljerte beskrivelsen som er gitt nedenfor.

[0007] Det følgende beskriver fremgangsmåter og systemer for å anvende visuelle effekter på aktivt innhold, så som knapper, kombinasjonsbokser, video, redigeringsfelter, etc., som bevarer interaktiviteten til det aktive innholdet. Det beskrives også en mekanisme som lar utviklere bygge opp nye visuelle effekter og anvende dem på aktivt innhold. Slike visuelle effekter omfatter for eksempel sløring, gløding, blinking, splittvisning, forstørring, krymping, uttoning, gråtoning, dreining, bølging, kombinerer, glitring eller en hvilken som helst annen synlig effekt som kan bli anvendt på aktivt innhold.

[0008] Mange av de ledsagende trekkene vil lettere innses etter hvert som de forstås bedre ved å henvise til den følgende detaljerte beskrivelsen, sett i forbindelse med de vedlagte figurene.

[0009] Foreliggende oppfinnelse er særlig egnet til å tilveiebringe en fremgangsmåte for å transformere koordinater for en treffhendelse, der fremgangsmåten omfatter det å:

tilveiebringe en rettet asyklisk graf som omfatter en første effekt og en siste effekt, en inngang og en utgang, den første effekten og den siste effekten inkluderer hver respektivt en første toveis transformasjonsfunksjon og en andre toveis transformasjonsfunksjon, hvor en inngang til den første effekten er koblet til inngangen til den rettede asykliske grafen, og hvor en utgang fra den siste effekten er koblet til utgangen til den rettede asykliske grafen;

fremvise et visuelt element på et område av en skjerm;

transformere det visuelle elementet ved å anvende den første toveis transformasjonsfunksjonen på det visuelle elementet som resulterer i et først-transformert visuelt element;

transformere det først-transformerte visuelle elementet ved å anvende den andre toveis transformasjonsfunksjonen på det først-transformerte visuelle elementet som resulterer i et sist-transformert visuelt element;

fremvise det sist-transformerte visuelle elementet i stedet for det visuelle elementet på skjermen;

ta imot koordinatene til treffhendelsen hvor treffhendelsen indikerer det fremviste sist-transformerte visuelle elementet;

5 transformere koordinatene ved å anvende den andre toveis transformasjonsfunksjonen på koordinatene i revers som resulterer i sist-transformerte koordinater;

transformere de sist-transformerte koordinatene ved å anvende den første toveis transformasjonsfunksjonen på de sist-transformerte koordinatene so  
10 resulterer i først-transformerte koordinater; og

bestemme at de først-transformerte koordinatene identifiserer et punkt i området, der fremgangsmåten er utført av an beregningsinnretning.

Foreliggende oppfinnelse er videre egnet til å tilveiebringe et system for å  
15 transformere koordinater til en treffhendelse, systemet innbefatter:

en datamaskininnretning som inkluderer en skjerm;

et effektobjekt som inkluderer en inngang og en utgang, der effektobjektet videre inkluderer en første effekt og en siste effekt som hver respektivt inkluderer en første toveis transformasjonsfunksjon og en andre toveis  
20 transformasjonsfunksjon, hvor en inngang til den første effekten er koblet til inngangen til effektobjektet, og hvor en utgang fra den første effekten er koblet til en inngang til den siste effekten, og hvor en utgang fra den siste effekten er koblet til utgangen til effektobjektet;

datamaskininnretningen er konfigurert til fremvise et visuelt element på et  
25 område av skjermen;

den første effekten er konfigurert til å transformere det visuelle elementet ved å anvende den første toveis transformasjonsfunksjonen på det visuelle elementet som resulterer i et først-transformert visuelt element;

den siste effekten er konfigurert til å transformere det først-transformerte visuelle elementet ved å anvende den andre toveis transformasjonsfunksjonen  
30 på det først-transformerte visuelle elementet som resulterer i et sist-transformert visuelt element;

datamaskininnretningen er videre konfigurert til å fremvise det sist-transformerte visuelle elementet i stedet for det visuelle elementet på skjermen;

datamaskininnretningen er videre konfigurert til å ta imot koordinatene til treffhendelsen hvor treffhendelsen indikerer det fremviste sist-transformerte visuelle elementet;

den siste effekten er videre konfigurert til å transformere koordinatene ved å anvende den andre toveis transformasjonsfunksjonen på koordinatene i revers som resulterer i sist-transformerte koordinater;

den første effekten er videre konfigurert til å transformere de sist-transformerte koordinatene ved å anvende den første toveis transformasjonsfunksjonen på de sist-transformerte koordinatene som resulterer i først-transformerte koordinater; og

datamaskininnretningen er videre konfigurert til å bestemme at de først-transformerte koordinatene identifiserer et punkt i området

## BESKRIVELSE AV FIGURENE

[0010] Foreliggende oppfinnelse vil forstås bedre fra den følgende detaljerte beskrivelsen, sett i lys av de vedlagte figurene, der:

[0011] Figur 1 er et blokkdiagram som viser et eksempel på effektobjekt (typen Effect).

[0012] Figur 2 er et blokkdiagram som viser et eksempel på effektobjekt som omfatter forskjellige innmatings- og utmatingskoblinger.

[0013] Figur 3 er et blokkdiagram som viser et eksempel på effektobjekt som inneholder to effekt-primitiver (typen EffectPrimitive), som støtter forskjellige inndata- og utdataformater.

[0014] Figur 4 er et blokkdiagram som viser et eksempel på effektobjekt.

[0015] Figur 5 er et blokkdiagram som viser et eksempel på knapp før og etter bruk av en eksplosjonseffekt.

[0016] Figur 6 er et blokkdiagram som viser et eksempel på fremgangsmåte for å tilveiebringe én eller flere effekter på et grafisk element.

[0017] Figur 7 er et blokkdiagram som viser et eksempel på fremgangsmåte for å transformere treffkoordinater for nøyaktig trefftesting på et grafisk element med én eller flere anvendte effekter.

[0018] Figur 8 er et blokkdiagram som viser et generelt databehandlingsmiljø som kan anvendes i én eller flere utførelser i henhold til beskrivelsen her.

[0019] Like referansenummer er anvendt for å angi like deler i de vedlagte figurene.

## 10 DETALJERT BESKRIVELSE

[0020] Den detaljerte beskrivelsen gitt nedenfor i forbindelse med de vedlagte figurene er ment som en beskrivelse av eksemplene gitt her, og er ikke ment å representere den eneste måten som eksemplene gitt her kan konstrueres eller anvendes. Beskrivelsen forklarer funksjonene og sekvensen av trinn for å konstruere og anvende eksempelet. Imidlertid kan de samme eller ekvivalente funksjoner og sekvenser oppnås av andre eksempler.

[0021] Selv om eksemplene gitt her er beskrevet og illustrert som realisert i et datasystem, er det beskrevne systemet ment som et eksempel og ikke en begrensning. Som fagmannen vil forstå er eksemplene gitt her egnet til bruk i en rekke forskjellige typer databehandlings- og grafikkssystemer.

[0022] Forskjellige betegnelser anvendt i denne beskrivelsen er eksplisitt definert nedenfor. De følgende definisjoner er ment å gå foran eventuelle alternative definisjoner som kan være kjent for fagmannen.

[0023] Aktivt innhold – et grafisk element, så som en knapp, en kombinasjonsboks, et redigeringsfelt, en avkrysningsboks, et bilde, en video eller liknende som er i stand til å støtte en eller annen form for brukervekselvirkning eller som er animert i en eller annen form. Aktivt innhold kan bli rendret i minne uten å bli vist på en skjerm.

[0024] Effect – Typen Effect er et objekt som inneholder en EffectGraph (dvs. en direkte asyklisk graf (DAG - Direct Acyclic Graph)).

[0025] EffectPrimitive – Dette er en grunneffekt. Den inneholder ingen underelementer. Det er denne effekten utviklere i alminnelighet oppretter. Dette er objektet som utfører den faktiske pikselbehandlingen for effekten.

[0026] effekt – Denne betegnelsen anvendes typisk når enten et Effect-objekt eller et EffectPrimitive-objekt kan anvendes eller spesifiseres.

[0027] EffectGraph – Denne refererer til hele effektgrafene (dvs. et Effect-objekt som er det øverste Effect-objektet i en DAG av et sett av DAG'er).

5 [0028] kobling – Denne anvendes for å beskrive koblingspunktet (via IMILBitmapEffectInputConnector) som en effekt eksponerer for å koble sammen effekter eller for andre formål.

[0029] Rastergrafikk – Dette er representasjon av bilder som en samling av piksler, generelt i form av et rektangulært nett eller punktgrafikk. Denne  
10 betegnelsen anvendes typisk motsatt av betegnelsen "vektorgrafikk".

[0030] Vektorgrafikk – Også kjent som geometrisk modellering, dette er bruk av geometriske primitiver, så som punkter, linjer, kurver og polygoner, til å representere bilder i datamaskingrafikk. Denne betegnelsen anvendes typisk  
motsatt av betegnelsen "rastergrafikk".

15 [0031] Grafisk element – Dette er en grafisk konstruksjon så som en knapp, et redigeringsfelt, et bilde, en meny, et ikon eller en hvilken som helst annen interaktiv eller animert grafisk konstruksjon som rendres i minne eller på en fremvisningsskjerm, et elektronisk lerret, i et brukergrensesnitt, eller liknende. Et grafisk element kan være et rastergrafikkelement, et  
20 vektorgrafikkelement eller en hvilken som helst annen form for grafisk konstruksjon.

[0032] Figur 1 er et blokkdiagram som viser et eksempel på Effect-objekt 100. Typen Effect er en samling av én eller flere EffectPrimitive-objekter, så som 104a-104d, som danner en direkte asyklisk graf (DAG). Denne effekt-  
25 DAG'en er typisk innkapslet i ett enkelt objekt 100 som eksponerer IMILBitmapEffect 106. Hvert element i Effect-objektet 100 er enten et annet Effect-objekt (dvs. implementerer IMILBitmapEffect) eller et EffectPrimitive-objekt (dvs. implementerer IMILBitmapEffectPrimitive 107). Hver effekt kan ha 0 eller flere innmatninger, så som innmatingen 108, og utmatninger, så som  
30 utmatingen 109. Hver inn/ut-kobling er beskrevet av en IMILBitmapEffectKonnector, så som eksemplene 101 og 102. Disse er koblet sammen, som eksemplifisert av forbindelsen 110, for å danne datastrømbanene gjennom effekt-DAG'en.

[0033] Effect-objektet 100 kan anvendes for å tilveiebringe en visuell effekt for aktivt innhold i punktgrafikkform, omfattende grafiske elementer så som knapper, redigeringsfelter, videografikk, eller hvilke som helst andre interaktive eller animerte grafiske elementer på en fremvisningsskjerm, et lerret, i et brukergrensesnitt, eller liknende. Effect-objektet 100 kan også brukes til å anvende en visuell effekt på punktgrafikk i minne uten at punktgrafikken noen gang blir vist. Videre kan Effect-objektet 100 eksponere andre egenskaper som beskriver attributter ved sin innmating. For eksempel kan et BitmapEffect-objekt sende ut fargehistograminformasjon eller flatedeteksjonsinformasjon. De spesifikke effektene som er tilgjengelig bestemmes i alminnelighet av Effect-objektene og EffectPrimitive-objektene inneholdt i et Effect-objekt, så som det eksemplifiserte Effect-objektet 100.

[0034] En punktgrafikkavbildning av et grafisk element blir typisk forsynt til innmatingen til det eksemplifiserte Effect-objektet 100, så som innmatingen 101. Den innmatede punktgrafikkavbildningen blir behandlet av Effect-objektet 100 for å frembringe en effekt (eller samling av effekter dersom Effect-objektet omfatter flere EffectPrimitive-objekter, så som 104a-104d, eller andre Effect-objekter), og en punktgrafikkavbildning som omfatter effekten(e) blir forsynt ved utmatingen fra Effect-objektet 100, så som utmatingen 102.

[0035] Figur 2 er et blokkdiagram som viser et eksempel på effektobjekt 200 som omfatter forskjellige inn- og utmatingskoblinger. Fordelen med å eksponere et sett av inn-/utmatingsskoblinger, så som koblingen 210, i stedet for kun å eksponere et utbilde er at slike koblinger gjør det mulig å spørre om effektens inn- og utdataformater og annen informasjon uten faktisk å kjøre effekten. Hver kobling kan motta/generere bilder i flere formater. Formatene en kobling mottar/genererer kan bli listet gjennom grensesnittene IMILBitmapEffectInputConnector eller IMILBitmapEffectOutputConnector sine metoder GetFormat() og GetNumberFormats(). Hvert koblingspunkt kan også ha et optimalt format — det formatet som effekten vil foretrekke å motta/generere som inndata/utdata. Dersom to sammenkoblede koblingspunkter har samme optimale format blir i alminnelighet dette formatet anvendt. Ellers bestemmes et annet sammenfallende format, typisk basert på det initiale utdataformatet ønsket i enden av effektgrafen. Dersom det ikke

finnes noe gjensidig sammenfallende format vil da en konvertering finne sted når data strømmer fra utmatingen fra én effekt til innmatingen til den neste effekten, som vist i forbindelse med figur 3.

5 [0036] Figur 3 er et blokkdiagram som viser et eksempel på effektobjekt 300 som inneholder to EffectPrimitive-objekter som støtter forskjellige inn- og utdataformater. EffectPrimitive1 er vist å støtte to utdataformater, MILPixelFormat24bppBGR – den optimale utmatingen som angitt av asterisken, og et andre utdataformat, MILPixelFormat32bppARGB, EffectPrimitive2 er vist å støtte to inndataformater, der det optimale inndataformatet er det samme som 10 det optimale utdataformatet til EffectPrimitive1. Koblingen 310 mellom EffectPrimitive1 og EffectPrimitive2 vil ikke forårsake noen formatkonvertering siden de har samme optimale formattypen MILPixelFormat24bppBGR. EffectPrimitive1 vil bli bedt om utmating i sitt optimale format, og resultatet vil bli sendt til EffectPrimitive2.

15 [0037] EffectPrimitive2 er vist å støtte ett enkelt utdataformat, MILPixelFormat32bppGray (dens optimale format som angitt av asterisken), og er koblet 320 til den ene utmatingen 330 fra effekten 300. Det eneste utdataformatet til effekten 300 er imidlertid MILPixelFormat32bppARGB, som er forskjellig fra det eneste utdataformatet til PrimitiveEffect2. Det vil derfor finne 20 sted en konvertering mellom utmatingen fra EffectPrimitive2 og den endelige utmatingen fra effektgrafens 300.

[0038] I ett eksempel på hvordan inkompatible formater håndteres, dersom de optimale formatene ikke sammenfaller for grafen og effekten koblet til utmatingen fra grafen, vil det da bli bedt om det siste instruerte formatet (dvs. 25 MILPixelFormat32bppGray som vist i figur 3). Dersom det siste instruerte utdataformatet ikke eksisterer som en utmating fra grafen, kan et tilfeldig format bli valgt.

[0039] Hver utmatingskobling (IMILBitmapEffectOutputConnector) kan være koblet til en innmating til en annen effekt i DAG'en, så lenge den 30 opprettholder grafens asykliske natur. Denne koblingen kan være 1-til-mange (dvs. at én enkelt utmatingskobling kan bli kjedet flere innmatingskoblinger, men ikke omvendt).

[0040] Figur 4 er et blokkdiagram som viser et eksempel på effektobjekt 400. I dette eksempelet består effektobjektet 400 av en innpakning 420 bygget opp ved anvendelse av styrt kode, så som C# (C sharp), som inneholder en ikke-styrt implementasjon av den faktiske effektkoden 410, skrevet i et språk så som C++ eller liknende. Det eksemplifiserte objektet 400 har én enkelt innmating 401 og én enkelt utmating 490. I alternative eksempler kan flere innmatinger og/eller utmatninger være tilveiebrakt. Dersom formatkonvertering er nødvendig eller tilveiebrakt, er konverteringskode typisk innlemmet i innpakningsobjektet 420. Alternativt kan konverteringskode være innlemmet i den ikke-styrte kodedelen 410 eller være tilveiebrakt av en annen modul.

[0041] Også innlemmet som del av effektobjektet 400 er en transformasjonsfunksjon  $F(p)$  430. I ett eksempel er transformasjonsfunksjonen 430 innlemmet som del av den ikke-styrte koden 410.

Transformasjonsfunksjonen 430 er den funksjonen som transformerer et innmatet grafisk element, typisk i form av en punktgrafikkavbildning, for å gi det grafiske elementet en ny visuell effekt. Generelt kan en slik transformasjonsfunksjon 430 anvendes på grafiske elementer som er basert på rastergrafikk, vektorgrafikk eller liknende.

[0042] Anta for eksempel at det er ønskelig å tilveiebringe en knapp i et brukergrensesnitt og at knappen "gløder" når musen plasseres over knappen. I et slikt tilfelle kan et glødeeffektobjekt bli opprettet og knyttet til knappen, og transformasjonsfunksjonen i glødeeffekten utfører operasjonen på punktgrafikkavbildningen av knappen for å gjøre at den synes å gløde. Merk at knappens funksjonalitet forblir uendret — bare de visuelle aspektene ved knappen blir modifisert av effekten. Nærmere bestemt kan brukeren fortsatt holde musemarkøren over knappen, klikke på knappen, etc., og knappen oppfører seg som forventet.

[0043] I et tradisjonelt brukergrensesnitt anvendes typisk "trefftesting" for å bestemme om en brukerinnmating, så som et museklikk eller et tastetrykk, skal anvendes på et grafisk element eller aktivt innhold. Nærmere bestemt, når brukeren for eksempel klikker på en knapp, sjekker trefftestkoden koordinatene til museklikket for å bestemme om klikket ble gjort på knappen. I så fall reagerer

knappen følgelig, for eksempel med å tegne seg på nytt i nedeposisjonen, iverksette funksjonaliteten forbundet med knappen, etc.

[0044] Dersom en visuell effekt er knyttet til knappen, er det viktig at trefftesting fortsatt fungerer som forventet. I eksempelet med et  
5 glødeeffektobjekt, når musemarkøren holdes over knappen, kan et tilhørende glødeeffektobjekt gjøre at knappen får et glødende utseende. I tillegg til glødeeffekten vil knappen også bli uthevet som følge av mus-over-hendelsen, som den normalt blir. Et klikk på knappen vil også gi det forventede resultatet. I ett eksempel kan et effektobjekt knyttes til et grafisk element som en egenskap i  
10 elementet.

[0045] Figur 5 er et blokkdiagram som viser et eksempel på knapp før og etter bruk av en eksplosjonseffekt. Betrakt et effektobjekt som genererer en eksplosjonseffekt — som deler opp det aktuelle grafiske elementet i flere fragmenter og sprer dem over et brukergrensesnitt over en tidsperiode. Figur 5  
15 viser et før-og-etter-eksempel på en slik effekt.

[0046] I ramme 501 — dvs. “før”-rammen, er brukergrensesnittet 500 vist med en tradisjonell knapp 502 før eksplosjonseffekten er anvendt, muligens som følge av en mus-over-hendelse eller en klikkehendelse på knappen. Ramme 502 — dvs. “etter”-rammen, viser samme brukergrensesnitt 500 med  
20 knappen 504 etter at eksplosjonseffekten er anvendt. Knappen 504 er vist delt i tre fragmenter 504a-504c, som er atskilt fra hverandre i brukergrensesnittet 500.

[0047] Eksplosjonseffekten implementeres delvis av en eksplosjon-transformasjonsfunksjon som avbilder hver bit i den innmatede punktgrafikken for  
25 knappen til nye posisjoner, for eksempel i flere fragmenter. Slike avbildninger kan endre seg over tid, for eksempel i et “beveger seg vekk”-aspekt ved en eksplosjon, muligens inntil fragmentene “faller til ro” et sted i brukergrensesnittet.

[0048] Uavhengig av transformasjonen fortsetter knappen å fungere som  
30 forventet. Nærmere bestemt kan brukeren klikke på et hvilket som helst fragment av den eksploderte (exploded button) knappen 504, og knappen vil oppføre seg som den ville dersom den ikke var oppdelt. Dette innebærer at trefftestkoden fortsetter å fungere som forventet. Dette lettes av

brukerhendelsene som går gjennom Effect-objektet og transformasjonsfunksjonen som bestemmer om knappen ble "truffet". Dette gjøres vanligvis ved å løse transformasjonsfunksjonen i revers, eller ved å anvende en toveis transformasjonsfunksjon.

5 [0049] Transformasjonsfunksjoner kan utføre affine transformasjoner eller ikke-affine transformasjoner. For eksempel kan affine transformasjoner omfatte forstørring, krymping, strekking eller rotering av punktgrafikk. Ikke-affine transformasjoner kan omfatte effekter så som sløring, gløding, deformasjon, gråtoning, lysdempning, uttoning, gnistring, gjenskin, blinking, omforming eller  
10 en hvilken som helst annen tilsvarende effekt, der elementer i punktgrafikken blir lagt til, fjernet og/eller modifisert på en eller annen måte. En transformasjonsfunksjon trenger videre ikke oversette alle de opprinnelige bitene i innmatet punktgrafikk, for eksempel når den frembringer effekter som forsvinning, krymping, utfalling, overgang til en linje eller et punkt, eller liknende.  
15 Andre mulige transformasjoner omfatter vridning, bølging, desintegring, duplisering, sammenslåing av flere punktgrafikkavbildninger til én, eller liknende. I noen tilfeller kan effekten føre til at en bit fra innmatet punktgrafikk blir avbildet til flere steder, til ett eller flere tilfeldige steder, eller fjernet helt. I alle de følgende eksemplene gjør transformasjonsfunksjonen det også mulig å  
20 angi om den modifiserte punktgrafikkavbildningen er "truffet" av en brukerinntasting for å sikre at treffestfunksjonaliteten fortsetter å fungere som forventet.

[0050] Figur 6 er et blokkdiagram som viser et eksempel på fremgangsmåte 600 for å tilveiebringe én eller flere effekter til et grafisk  
25 element. Et Effect-objekt, så som Effect-objektet 100 som er vist i figur 1 og anvendes i beskrivelsen av fremgangsmåten 600 nedenfor, kan anvende en slik fremgangsmåte 600.

[0051] I trinn 610 begynner fremgangsmåten 600 med en punktgrafikkavbildning av et grafisk element tilveiebrakt ved inntastingen til  
30 Effect-objektet 100. Dette kan skje gjennom en rekke tradisjonelle programvareprosesser og/eller metoder.

[0052] I trinn 620 bestemmer en test om det finnes et neste EffectPrimitive-objekt inneholdt i Effect-objektet 100 sin DAG for å behandle

punktgrafikkavbildningen. Som vist i figur 1 kan et Effect-objekt inneholde EffectPrimitive-objekter og/eller andre Effect-objekter der én eller flere av innmatingene er koblet til innmatingen til Effect-objektet 100 og hver utmating er koblet til ett eller flere ytterligere innmatinger til EffectPrimitive-objekter og/eller Effect-objekter eller til utmatingen fra Effect-objektet 100. Dersom det finnes et neste EffectPrimitive-objekt, fortsetter fremgangsmåten 600 i trinn 650. Ellers fortsetter fremgangsmåten 600 i trinn 630.

[0053] I trinn 630 bestemmer en test om et neste Effect-objekt er inneholdt i Effect-objektets 100 DAG. Dersom det finnes et neste Effect-objekt, for eksempel "Effect-objekt B", fortsetter fremgangsmåten i trinn 610 for Effect-objekt B. Ellers fortsetter fremgangsmåten 600 i trinn 640.

[0054] I trinn 640 er den innmatede punktgrafikkavbildningen ferdig behandlet av Effect-objektet 100, og den behandlede punktgrafikkavbildningen blir forsynt til utmatingen. Dette kan skje gjennom en rekke mulige tradisjonelle programvareprosesser og/eller metoder.

[0055] I trinn 650 blir punktgrafikkavbildningen, som kan ha blitt behandlet av eventuelle foregående EffectPrimitive-objekter og/eller Effect-objekter, sendt til innmatingen til det neste EffectPrimitive-objektet. Dette kan skje gjennom en rekke mulige tradisjonelle programvare prosesser og/eller metoder.

[0056] I trinn 660 bestemmer en test om punktgrafikkavbildningen er i et akseptabelt format for behandling av EffectPrimitive-objektet. I så fall fortsetter fremgangsmåten 600 i trinn 680. Ellers fortsetter fremgangsmåten 600 i trinn 670.

[0057] I trinn 670 blir punktgrafikkavbildningen konvertert eller omformatert til et akseptabelt format for behandling av det neste EffectPrimitive-objektet. I ett eksempel kan punktgrafikkavbildningen bli omformatert til et optimalt format.

[0058] I trinn 680 blir en transformasjonsfunksjon, så som transformasjonsfunksjonen  $F(p)$  430 vist i figur 4, anvendt på den innmatede punktgrafikkavbildningen, noe som typisk resulterer i at én eller flere effekter blir anvendt på punktgrafikkavbildningen eller at en ny punktgrafikkavbildning blir generert fra den innmatede punktgrafikkavbildningen, der den nye

punktgrafikkavbildningen omfatter effekten(e). Fremgangsmåten 600 fortsetter i trinn 620.

[0059] Figur 7 er et blokkdiagram som viser et eksempel på fremgangsmåte 700 for å transformere treffkoordinater for nøyaktig trefftesting på et grafisk element med én eller flere anvendte effekter. Fordi et grafisk element med én eller flere effekter kan befinne seg i et annet brukergrensesnitt enn det ville uten effekten(e), er typisk transformasjon av treffkoordinater, så som en museklikk- eller mus-over-hendelse nødvendig for å sikre korrekt trefftesting. Et Effect-objekt, så som Effect-objektet 100 vist i figur 1 og anvendt i beskrivelsen av fremgangsmåten 700 nedenfor, kan anvende en slik fremgangsmåte 700.

[0060] I trinn 710 begynner fremgangsmåten 700 med at koordinatene for en brukerhendelse, så som en museklikk- eller mus-over-hendelse, forsynes til en innmating til Effect-objektet 100. Dette kan skje gjennom en rekke mulige tradisjonelle programvareprosesser og/eller metoder.

[0061] I trinn 720 bestemmer en test om det finnes et foregående EffectPrimitive-objekt inneholdt i Effect-objektets 100 DAG for å transformere koordinatene. Som vist i figur 1 kan et Effect-objekt, så som det eksemplifiserte Effect-objektet 100, inneholde EffectPrimitive-objekter og/eller andre Effect-objekter der én eller flere av innmatingene er koblet til innmatingen til Effect-objektet 100 og hver utmating er koblet til én eller flere innmatinger til ytterligere EffectPrimitive- og/eller Effect-objekter, eller til utmatingen fra Effect-objektet 100. Fremgangsmåten 700 begynner med det siste slike effektobjektet i DAG-grafen og jobber seg forover for å bevirke koordinattransformasjon. Et "foregående" effektobjekt er et effektobjekt i DAG-grafen med start ved utmatingen fra Effect-objekt og videre bakover gjennom alle Effect-objekter i DAG-grafen mot innmatingen til Effect-objektet. For eksempel er EffectPrimitive-objektet 104d foregående til Effect-objektet 100 sin utmating 102, EffectPrimitive-objektene 104a, 104b og 104c er foregående til 104d, og EffectPrimitive-objektet 104b er også foregående til 104c. Dersom det finnes et foregående EffectPrimitive-objekt, fortsetter fremgangsmåten 700 i trinn 730. Ellers fortsetter fremgangsmåten 700 i trinn 740.

[0062] I trinn 730 blir en transformasjonsfunksjon, så som transformasjonsfunksjonen  $F(p)$  430 vist i figur 4, anvendt på koordinatene til brukerens punktgrafikkavbildning. Transformasjonsfunksjonen blir typisk anvendt i revers. Fremgangsmåten 700 fortsetter i trinn 720.

5 [0063] I trinn 740 bestemmer en test om det finnes et foregående Effect-objekt inneholdt i Effect-objektets 100 DAG-graf. Dersom det finnes et foregående Effect-objekt, f.eks. "Effect-objekt A", fortsetter fremgangsmåten i trinn 710 for Effect-objekt A. Ellers fortsetter fremgangsmåten 700 i trinn 750.

10 [0064] I trinn 750 forsyner fremgangsmåten 700 de transformerte koordinatene for brukerhendelsen til trefftestkode eller liknende.

[0065] Figur 8 er et blokkdiagram som viser et generelt databehandlingsmiljø 800 som kan anvendes i én eller flere utførelser i henhold til beskrivelsen gitt her. Datasystemmiljøet 800 er bare ett eksempel på et passende databehandlingsmiljø, og er ikke ment å antyde noen som helst  
15 begrensning når det gjelder bruksområdet eller funksjonaliteten til oppfinnelsen det kreves beskyttelse for. Heller ikke skal databehandlingsmiljøet 800 tolkes som å ha noen som helst avhengighet eller krav vedrørende noen som helst enkeltkomponent eller kombinasjon av komponenter illustrert i det eksemplifiserte kjøremiljøet 800.

20 [0066] De beskrevne fremgangsmåter og objekter kan anvendes med en rekke forskjellige andre generelle eller spesialiserte datasystemmiljøer eller -anordninger. Eksempler på velkjente datasystemer, -miljøer og/eller -anordninger som kan være egnet til bruk omfatter, men er ikke begrenset til personlige datamaskiner, tjenermaskiner, håndholdte anordninger eller bærbare  
25 datamaskinanordninger, flerprosessorsystemer, mikroprosessorbaserte systemer, dekodebokser, programmerbar forbrukerelektronikk, personlige datamaskiner tilknyttet nettverk, minidatamaskiner, stormaskiner, distribuerte databehandlingsmiljøer som omfatter hvilke som helst av de ovenfor angitte systemer eller anordninger, og liknende.

30 [0067] Den følgende beskrivelsen kan gis i den generelle sammenhengen datamaskin-utførbare instruksjoner, så som programmoduler, som eksekveres av en datamaskin. Generelt omfatter programmoduler rutiner, programmer, objekter, komponenter, datastrukturer, etc. som utfører bestemte

oppgaver eller implementerer bestemte abstrakte datatyper. De beskrevne utførelsene kan også praktiseres i distribuerte databehandlingsmiljøer der oppgaver blir utført av fjerne prosesseringsanordninger som er sammenkoblet gjennom et kommunikasjonsnettverk. I et distribuert databehandlingsmiljø kan

5 programmoduler befinne seg i både lokale og fjerne datalagringsmedier, omfattende minnelagringsanordninger.

[0068] Med henvisning til figur 8 omfatter et eksempel på system for å realisere oppfinnelsen en generell databehandlingsanordning i form av en datamaskin 810. Komponenter i datamaskinen 810 kan omfatte, men er ikke

10 begrenset til en prosesseringsenhet 820, et systemminne 830 og en systembuss 821 som kobler forskjellige systemkomponenter omfattende systemminnet til prosesseringsenheten 820. Systembussen 821 kan være en hvilken som helst av forskjellige typer busstrukturer, omfattende en minnebuss eller minnestyringsenhet, en ekstern buss og en lokal buss, som anvender en

15 hvilken som helst av en rekke forskjellige bussarkitekturer. Som et eksempel, og ikke en begrensning omfatter slike arkitekturer en ISA-(Industry Standard Architecture)-buss, en MCA-(Micro Channel Architecture)-buss, en Enhanced ISA (EISA) buss, en lokal VESA-(Video Electronics Standards Association)-buss og en PCI-(Peripheral Component Interconnect)-buss, også kjent som en

20 Mezzanine-buss.

[0069] Datamaskinen 810 omfatter typisk en rekke forskjellige datamaskinlesbare medier. Datamaskinlesbare medier kan være hvilke som helst tilgjengelige medier som kan aksesserer av datamaskinen 810, og omfatter både volatile og ikke-volatile medier samt flyttbare og stasjonære

25 medier. Som et eksempel, og ikke en begrensning kan datamaskinlesbare medier omfatte datalagringsmedier og kommunikasjonsmedier. Datalagringsmedier omfatter volatile og ikke-volatile, flyttbare og stasjonære medier realisert med en hvilken som helst metode eller teknologi for lagring av informasjon, så som datamaskinlesbare instruksjoner, datastrukturer,

30 programmoduler eller andre data. Datalagringsmedier omfatter, men er ikke begrenset til RAM, ROM, EEPROM, flashminne eller annen minneteknologi, CD-ROM, DVD eller andre optiske platelagre, magnetkassetter, magnetbånd, magnetplatelagre eller andre magnetiske lagringsanordninger, eller et hvilket

som helst annet medium som kan anvendes for å lagre den ønskede informasjonen og som kan aksesseres av datamaskinen 810.

Kommunikasjonsmedier inneholder typisk datamaskinlesbare instruksjoner, datastrukturer og/eller programmer. Enhver kombinasjon av hvilke som helst av det ovennevnte skal også være omfattet innenfor rammen av datamaskinlesbare medier.

[0070] Systemminnet 830 omfatter datalagringsmedier i form av volatil og/eller ikke-volatil minne, så som leseminne (ROM) 831 og direkteaksessminne (RAM) 832. Et grunnleggende inn/ut-system 833 (BIOS), som inneholder de grunnleggende rutinene som bidrar til å overføre informasjon mellom elementer i datamaskinen 810, for eksempel under oppstart, er typisk lagret i ROM 831. RAM 832 inneholder typisk data og/eller programmoduler som er umiddelbart tilgjengelige for og/eller jobbes med av prosesseringsenheten 820. Som et eksempel, og ikke en begrensning illustrerer figur 8 et operativsystem 834, applikasjonsprogrammer 835, andre programmoduler 836 og programdata 837.

[0071] Datamaskinen 810 kan også omfatte andre flyttbare/stasjonære, volatile/ikke-volatile datalagringsmedier. Kun som et eksempel illustrerer figur 8 et harddiskdrev 841 som leser fra eller skriver til stasjonære, ikke-volatile magnetiske medier, et magnetplatedrev 851 som leser fra eller skriver til et flyttbart, ikke-volatil magnetplatelager 852, og et optisk platedrev 855 som leser fra eller skriver til et flyttbart, ikke-volatil optisk platelager 856, så som et CD-ROM eller annet optisk medium. Andre flyttbare/stasjonære, volatile/ikke-volatile datalagringsmedier som kan anvendes i det eksemplifiserte kjøremiljøet omfatter, men er ikke begrenset til magnetbånd/-kassetter, flashminnekort, DVD, digitalt videobånd, monolittisk RAM (solid state RAM), monolittisk ROM (solid state ROM) og liknende. Harddiskdrevet 841 er typisk koblet til systembussen 821 gjennom et grensesnitt for stasjonært minne, så som grensesnittet 840, og magnetplatedrevet 851 og det optiske platedrevet 855 er typisk koblet til systembussen 821 via et grensesnitt for flyttbart minne, så som grensesnittet 850.

[0072] Drevene og deres tilhørende datalagringsmedier beskrevet over og illustrert i figur 8 muliggjør lagring av datamaskinlesbare instruksjoner,

datastrukturer, programmoduler og andre data for datamaskinen 810. I figur 8 er for eksempel harddiskdrevet 841 illustrert som lagrende et operativsystem 844, applikasjonsprogrammer 845, andre programmoduler 846 og programdata 847. Merk at disse komponentene enten kan være de samme som eller

5 forskjellige fra operativsystem 834, applikasjonsprogrammer 835, andre programmoduler 836 og programdata 837. Operativsystem 844, applikasjonsprogrammer 845, andre programmoduler 846 og programdata 847 er gitt forskjellige referansenummer her for å illustrere det at de i det minste er forskjellige kopier. En bruker kan mate inn kommandoer og informasjon til

10 datamaskinen 810 gjennom innmatingsanordninger, så som et tastatur 862 og en pekeranordning 861, vanligvis omfattende en mus, styrekule eller berøringsmatte. Andre innmatingsanordninger (ikke vist) kan omfatte en mikrofon, styrespak, spillkontroll, parabolantenne, skanner eller liknende. Disse og andre innmatingsanordninger er ofte koblet til prosesseringsenheten 820

15 gjennom et brukerinntastingsgrensesnitt 860 som er koblet til systembussen 821, men kan også være tilkoblet via andre grensesnitt og busstrukturer, så som en parallellport, spillport eller USB-port. En dataskjerm 891 eller en annen type fremvisningsanordning er også koblet til systembussen 821 via et grensesnitt, så som et skjermgrensesnitt 890. I tillegg til dataskjermen kan

20 datamaskiner også omfatte andre eksterne utmatingsanordninger, så som høyttalere 897 og en skriver 896, som kan være tilkoblet gjennom et grensesnitt 895 for utmatning til eksterne enheter. Et kamera 863 (så som et digitalt/elektronisk stillbilde- eller videokamera, eller en film-/fotografiskanner) som er i stand til å ta en sekvens av bilder 864, kan også være omfattet som en

25 innmatingsanordning til den personlige datamaskinen 810. Selv om kun ett kamera er vist vil også flere kameraer kunne være omfattet som innmatingsanordninger til den personlige datamaskinen 810. Bildene 864 fra det ene eller de flere kameraene blir matet inn til datamaskinen 810 gjennom et passende kameragrensesnitt 865. Dette grensesnittet 865 er koblet til system-

30 bussen 821, slik at bildene kan bli sendt til og lagret i RAM 832 eller en av de andre datalagringsanordningene tilknyttet datamaskinen 810. Det skal imidlertid bemerkes at bilde-data også kan bli matet inn til datamaskinen 810 fra et hvilken

som helst av ovennevnte datamaskinlesbare mediene uten å kreve bruk av kameraet 863.

[0073] Datamaskinen 810 kan kjøre i et nettverksmiljø som anvender logiske forbindelser til én eller flere fjerne datamaskiner, så som en fjern datamaskin 880. Den fjerne datamaskinen 880 kan være en personlig datamaskin, en tjener, en ruter, en nettverkstilknyttet PC, en peer-anordning eller en annen vanlig nettverksnode, og omfatter typisk mange av eller alle elementene beskrevet over i forbindelse med datamaskinen 810, selv om bare en minnelagringsanordning 881 er illustrert i figur 8. De logiske forbindelsene vist i figur 8 omfatter et lokalt nettverk (LAN) 871 og et regionalt nettverk (WAN) 873, men kan også omfatte andre nettverk. Slike nettverksmiljøer er vanlige i kontorer, bedriftsdatanettverk, intranett og på Internett.

[0074] Når den anvendes i et LAN-miljø, er datamaskinen 810 koblet til LAN 871 gjennom et nettverksgrensesnitt eller -adapter 870. Når den anvendes i et WAN-miljø, omfatter datamaskinen 810 typisk et modem 872 eller en annen anordning for å etablere kommunikasjon over WAN 873, for eksempel Internett. Modemet 872, som kan være internt eller eksternt, kan være koblet til systembussen 821 via brukerinmatingsgrensesnittet 860 eller en annen passende mekanisme. I et nettverksmiljø kan programmoduler vist i datamaskinen 810, eller deler av disse, være lagret i den fjerne minnelagringsanordningen. Som et eksempel, og ikke en begrensning illustrerer figur 8 fjerne applikasjonsprogrammer 885 som lagret på minneanordningen 881. Det vil forstås at de viste nettverksforbindelsene kun er eksempler, og at andre anordninger for å opprette en kommunikasjonsforbindelse mellom datamaskinene kan anvendes.

[0075] Selv om ett eller flere utførelseseksempler er illustrert og beskrevet her, vil det forstås at en kan gjøre forskjellige endringer i disse uten å fjerne seg fra de vedføyde kravenes ramme og idé.

## PATENTKRAV

1. Fremgangsmåte for å transformere koordinater for en treffhendelse, der fremgangsmåten omfatter det å:

5 tilveiebringe en rettet asyklisk graf som omfatter en første effekt og en siste effekt, en inngang og en utgang, den første effekten og den siste effekten inkluderer hver respektivt en første toveis transformasjonsfunksjon og en andre toveis transformasjonsfunksjon, hvor en inngang til den første effekten er koblet til inngangen til den rettede asykliske grafen, og hvor en utgang fra den siste  
10 effekten er koblet til utgangen til den rettede asykliske grafen;

fremvise et visuelt element på et område av en skjerm;

transformere det visuelle elementet ved å anvende den første toveis transformasjonsfunksjonen på det visuelle elementet som resulterer i et første-transformert visuelt element;

15 transformere det først-transformerte visuelle elementet ved å anvende den andre toveis transformasjonsfunksjonen på det først-transformerte visuelle elementet som resulterer i et sist-transformert visuelt element;

fremvise det sist-transformerte visuelle elementet i stedet for det visuelle elementet på skjermen;

20 ta imot koordinatene til treffhendelsen hvor treffhendelsen indikerer det fremviste sist-transformerte visuelle elementet;

transformere koordinatene ved å anvende den andre toveis transformasjonsfunksjonen på koordinatene i revers som resulterer i sist-transformerte koordinater;

25 transformere de sist-transformerte koordinatene ved å anvende den første toveis transformasjonsfunksjonen på de sist-transformerte koordinatene so resulterer i først-transformerte koordinater; og

bestemme at de først-transformerte koordinatene identifiserer et punkt i området, der fremgangsmåten er utført av an beregningsinnretning.

30

2. Fremgangsmåte ifølge krav 1, der det grafiske elementet er en punkt-grafikkavbildning.

3. Fremgangsmåte ifølge krav 1, der det grafiske elementet inkluderer video.

5 4. Fremgangsmåte ifølge krav 1, videre innbefatter konvertering av det grafiske elementet til en punktgrafikkavbildning.

5. Fremgangsmåte ifølge krav 1, der det grafiske elementet animeres.

10 6. Lagringsmedia for datamaskin innbefattende datamaskinutførbare instruksjoner som, når de blir utført i en datamaskininnretning, forårsaker at datamaskininnretningen utfører en fremgangsmåte for å transformere koordinater til en treffhendelse, der fremgangsmåten innbefatter:

15 tilveiebringe en rettet asyklisk graf som omfatter en første effekt og en siste effekt, en inngang og en utgang, den første effekten og den siste effekten inkluderer hver respektivt en første toveis transformasjonsfunksjon og en andre toveis transformasjonsfunksjon, hvor en inngang til den første effekten er koblet til inngangen til den rettede asykliske grafen, og hvor en utgang fra den siste effekten er koblet til utgangen til den rettede asykliske grafen;

fremvise et visuelt element på et område av en skjerm;

20 transformere det visuelle elementet ved å anvende den første toveis transformasjonsfunksjonen på det visuelle elementet som resulterer i et først-transformert visuelt element;

25 transformere det først-transformerte visuelle elementet ved å anvende den andre toveis transformasjonsfunksjonen på det først-transformerte visuelle elementet som resulterer i et sist-transformert visuelt element;

fremvise det sist-transformerte visuelle elementet i stedet for det visuelle elementet på skjermen;

ta imot koordinatene til treffhendelsen hvor treffhendelsen indikerer det fremviste sist-transformerte visuelle elementet;

30 transformere koordinatene ved å anvende den andre toveis transformasjonsfunksjonen på koordinatene i revers som resulterer i sist-transformerte koordinater;

transformere de sist-transformerte koordinatene ved å anvende den første toveis transformasjonsfunksjonen på de sist-transformerte koordinatene som resulterer i først-transformerte koordinater; og

5 bestemme at de først-transformerte koordinatene identifiserer et punkt i området, der fremgangsmåten er utført av en beregningsinnretning.

7. Fremgangsmåte ifølge krav 1, der det grafiske elementet er en punkt-grafikkavbildning.

10 8. Fremgangsmåte ifølge krav 1, der det grafiske elementet inkluderer video.

9. Fremgangsmåte ifølge krav 1, der det grafiske elementet animeres.

15 10. System for å transformere koordinater til en treffhendelse, systemet innbefatter:

en datamaskininnretning som inkluderer en skjerm;

20 et effektobjekt som inkluderer en inngang og en utgang, der effektobjektet videre inkluderer en første effekt og en siste effekt som hver respektivt inkluderer en første toveis transformasjonsfunksjon og en andre toveis transformasjonsfunksjon, hvor en inngang til den første effekten er koblet til inngangen til effektobjektet, og hvor en utgang fra den første effekten er koblet til en inngang til den siste effekten, og hvor en utgang fra den siste effekten er koblet til utgangen til effektobjektet;

25 datamaskininnretningen er konfigurert til fremvise et visuelt element på et område av skjermen;

den første effekten er konfigurert til å transformere det visuelle elementet ved å anvende den første toveis transformasjonsfunksjonen på det visuelle elementet som resulterer i et først-transformert visuelt element;

30 den siste effekten er konfigurert til å transformere det først-transformerte visuelle elementet ved å anvende den andre toveis transformasjonsfunksjonen på det først-transformerte visuelle elementet som resulterer i et sist-transformert visuelt element;

datamaskininnretningen er videre konfigurert til å fremvise det sist-transformerte visuelle elementet i stedet for det visuelle elementet på skjermen;

datamaskininnretningen er videre konfigurert til å ta imot koordinatene til treffhendelsen hvor treffhendelsen indikerer det fremviste sist-transformerte visuelle elementet;

den siste effekten er videre konfigurert til å transformere koordinatene ved å anvende den andre toveis transformasjonsfunksjonen på koordinatene i revers som resulterer i sist-transformerte koordinater;

den første effekten er videre konfigurert til å transformere de sist-transformerte koordinatene ved å anvende den første toveis transformasjonsfunksjonen på de sist-transformerte koordinatene som resulterer i først-transformerte koordinater; og

datamaskininnretningen er videre konfigurert til å bestemme at de først-transformerte koordinatene identifiserer et punkt i området.

11. System ifølge krav 10 hvor det visuelle elementet er en punkt-grafikkavbildning.

12. System ifølge krav 10, hvor det visuelle elementet animeres.

13. System ifølge krav 10, hvor det visuelle elementet inkluderer video.

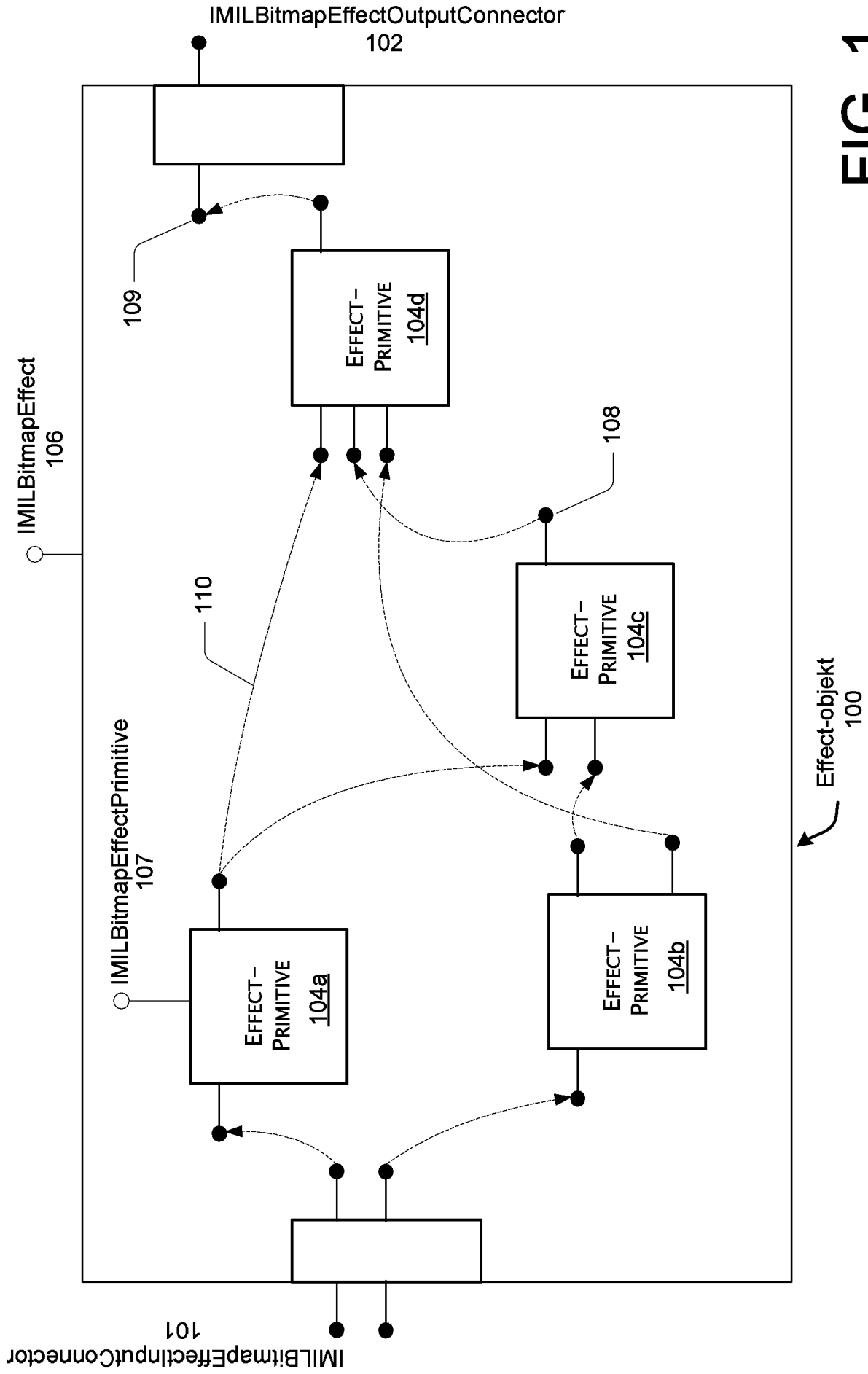


FIG. 1

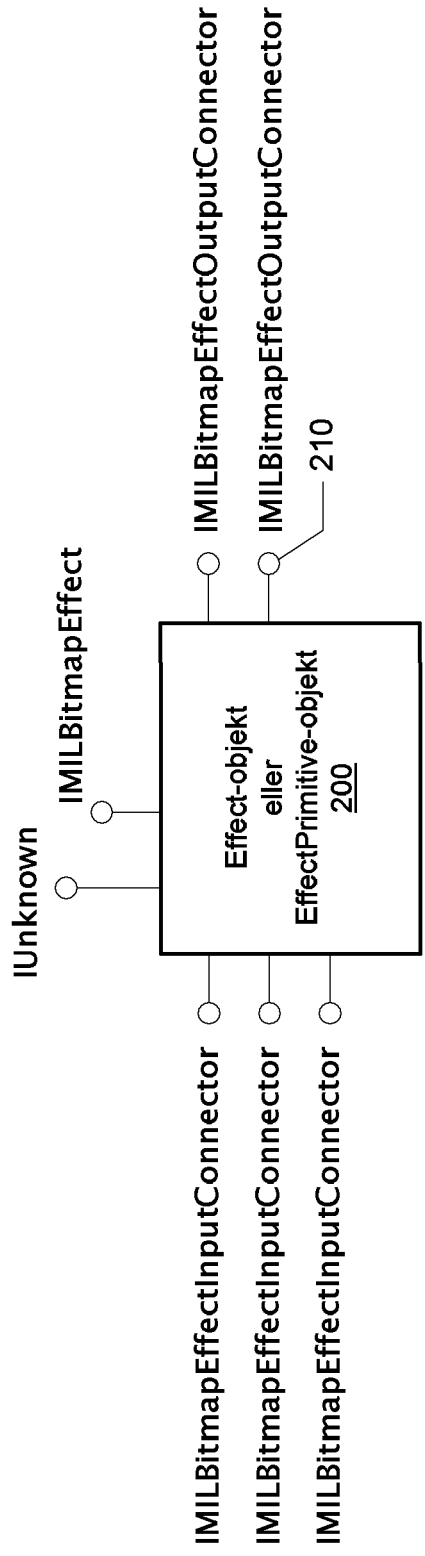


FIG. 2

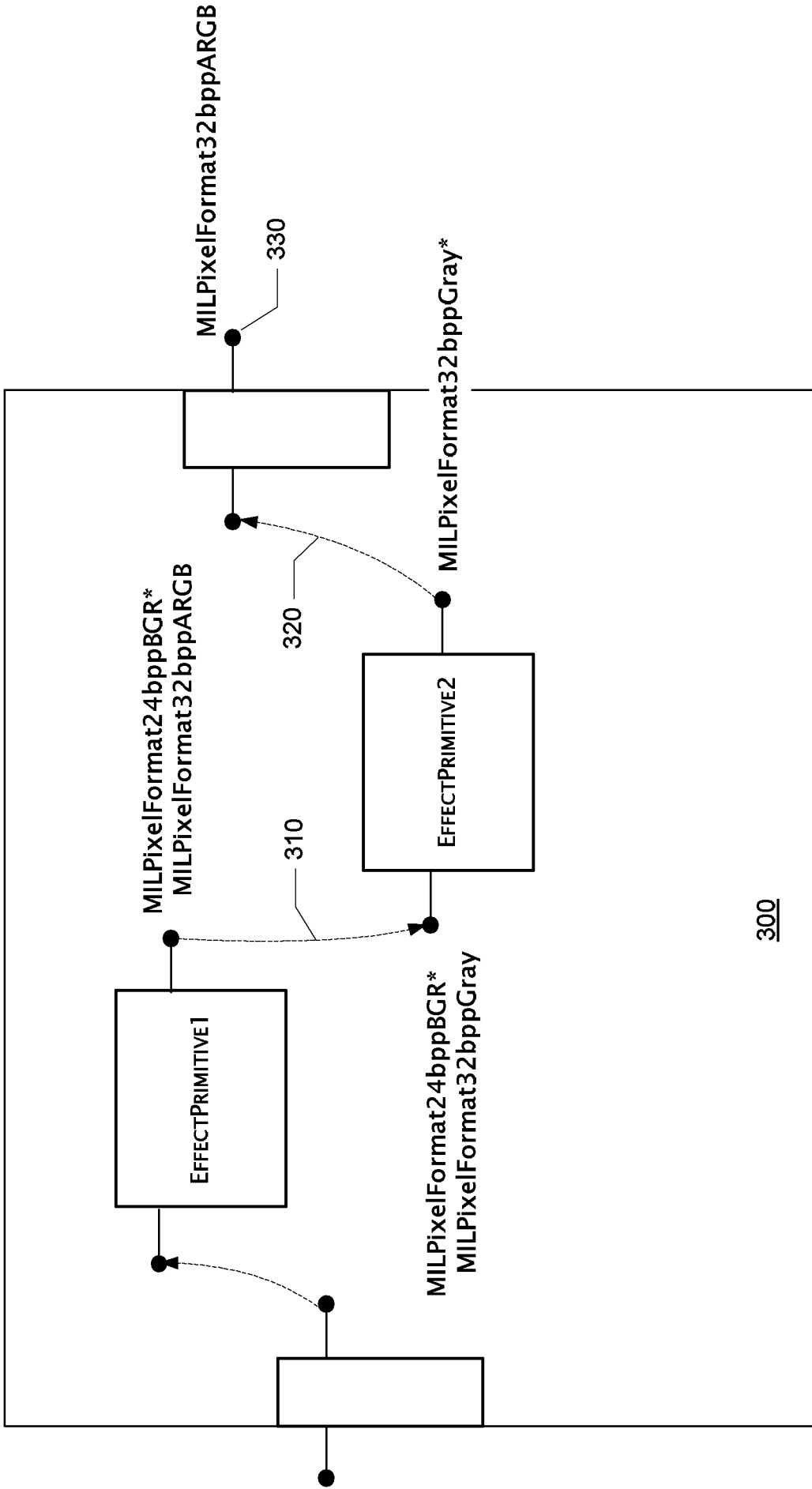


FIG. 3

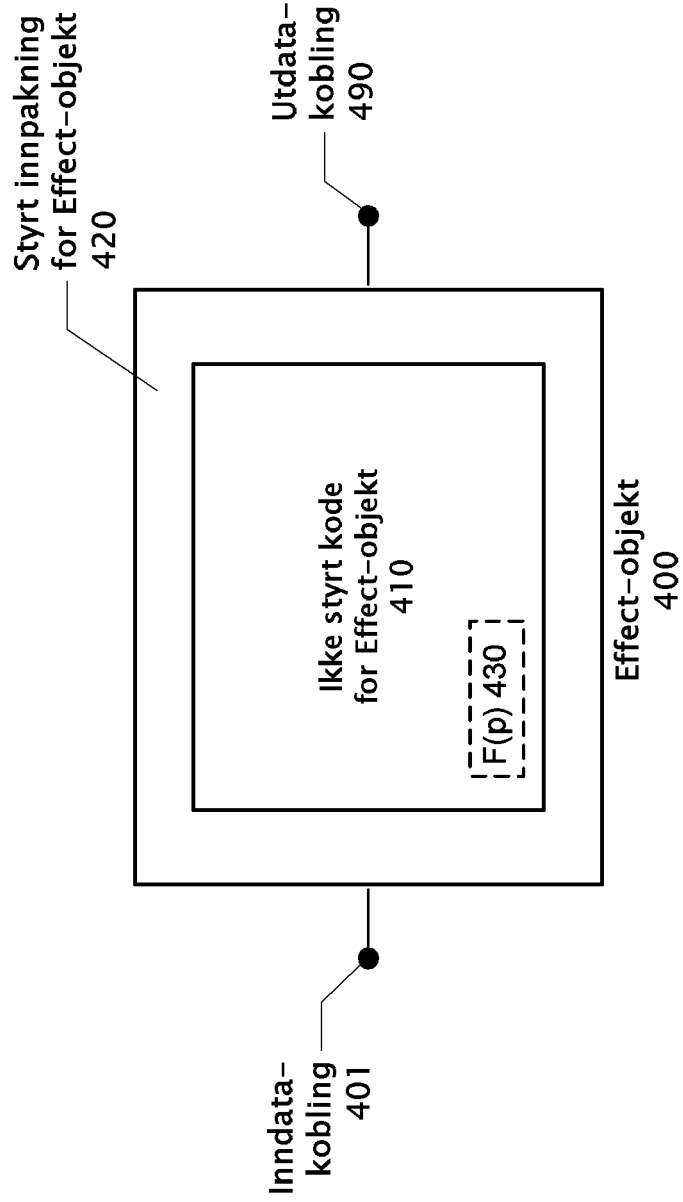
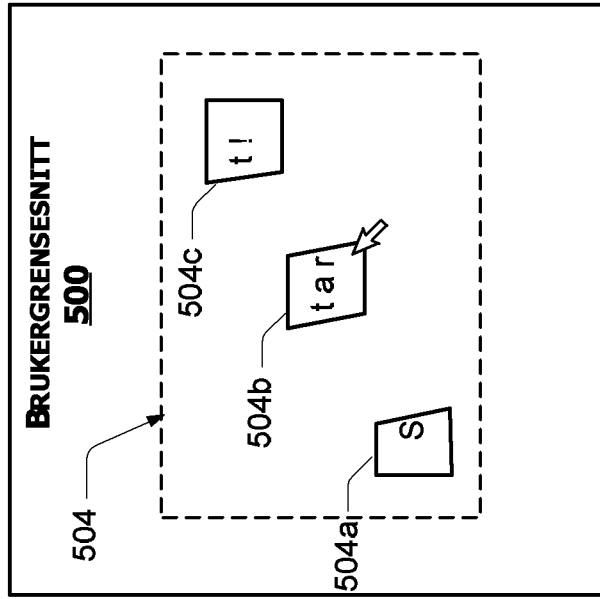
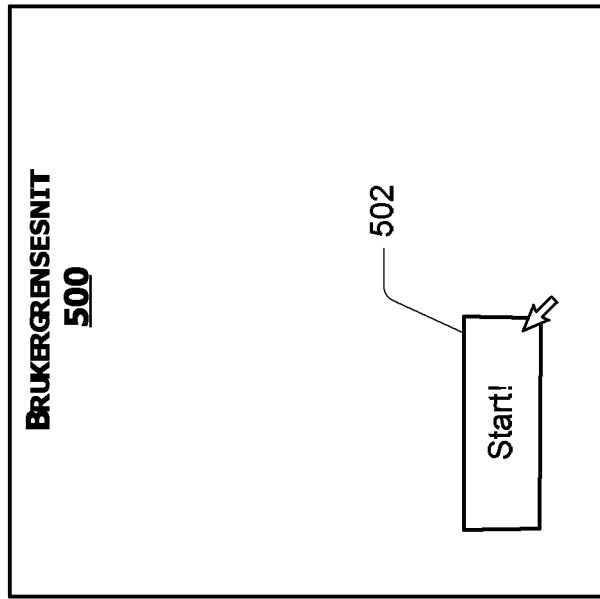
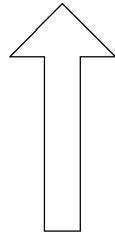


FIG. 4



502 Etter:  
Knapp forblir aktiv etter bruk  
av effekten



501 Før:  
Gr. snitt med aktiv  
knapp før bruk av  
effekt

FIG. 5

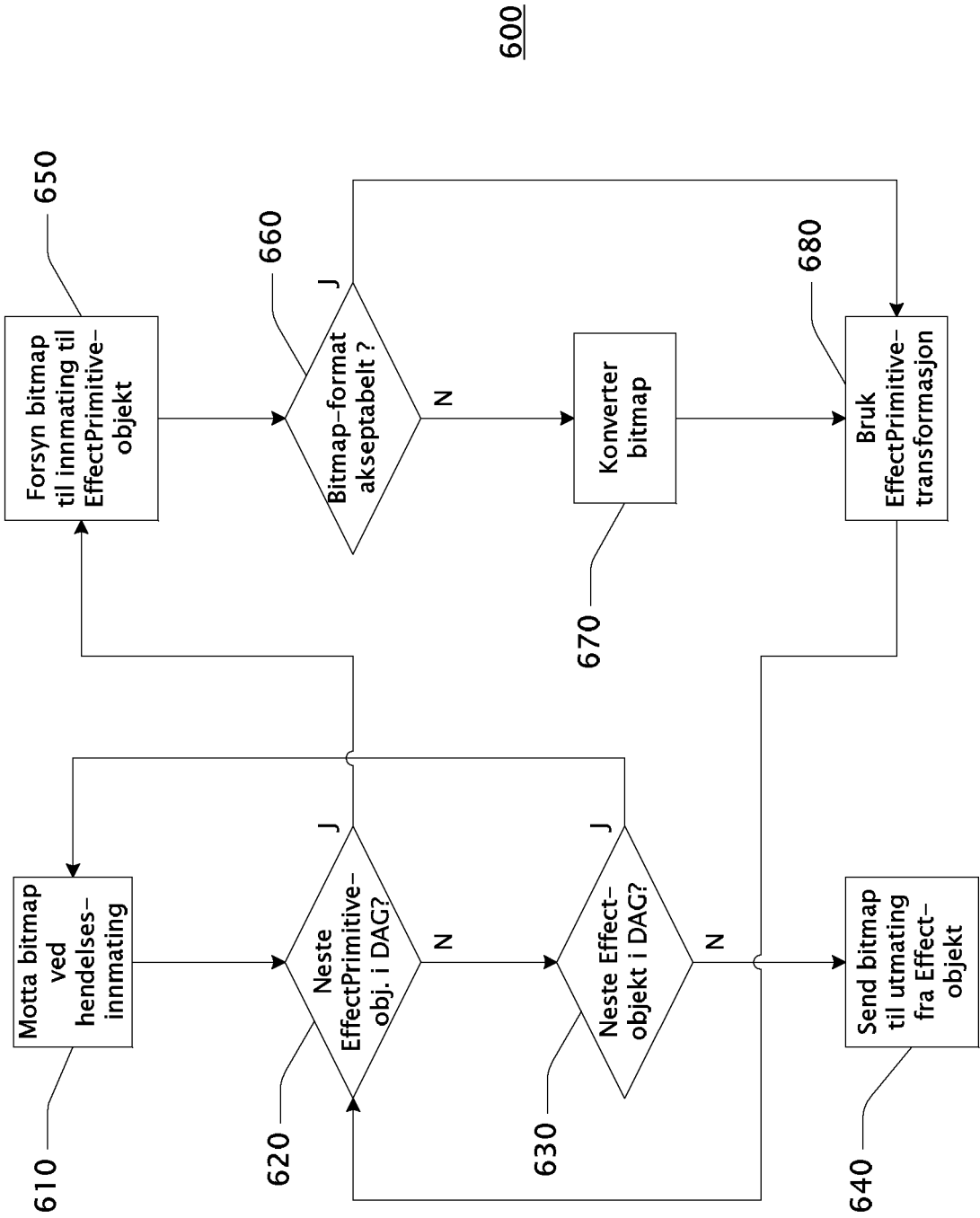


FIG. 6

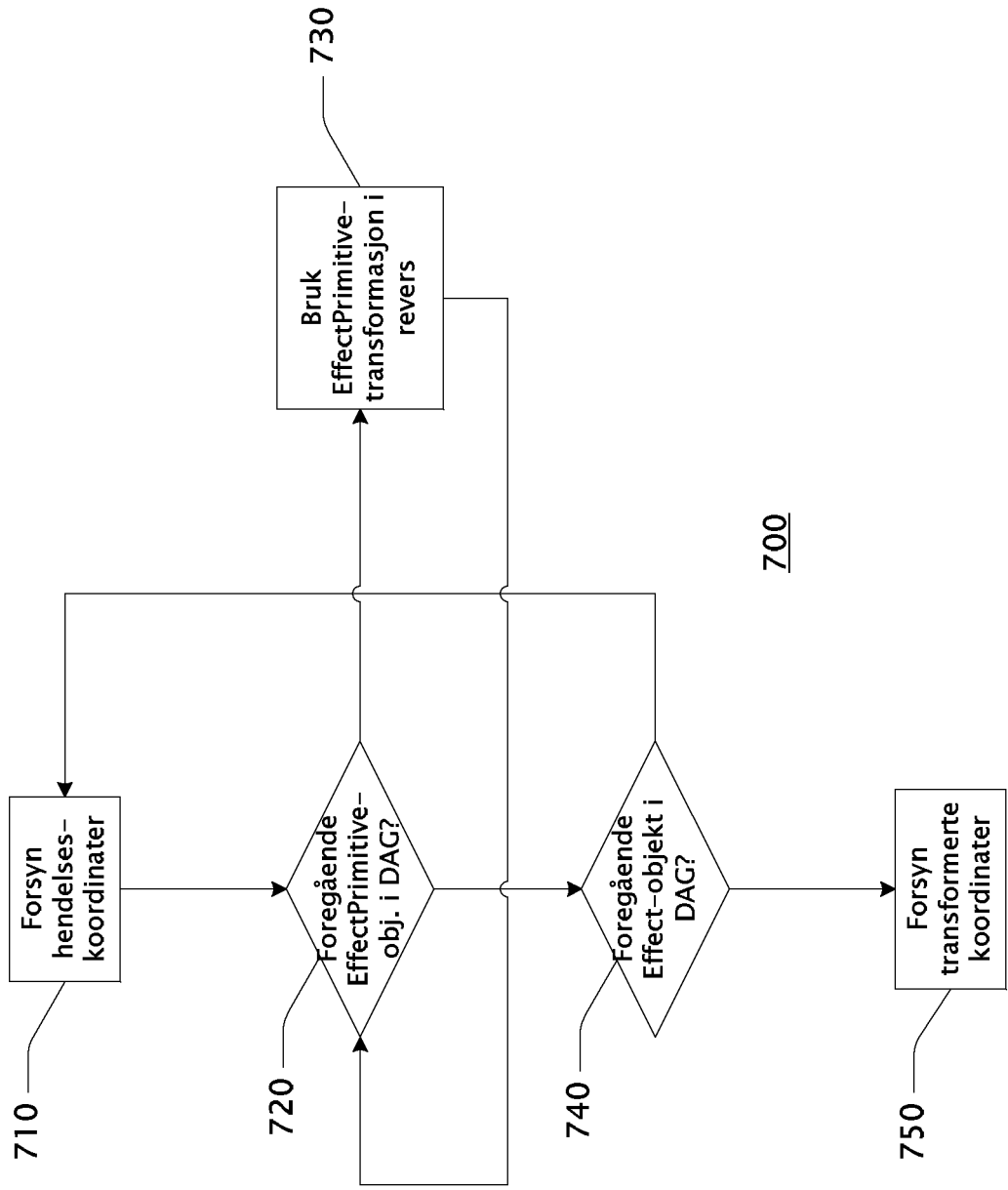


FIG. 7

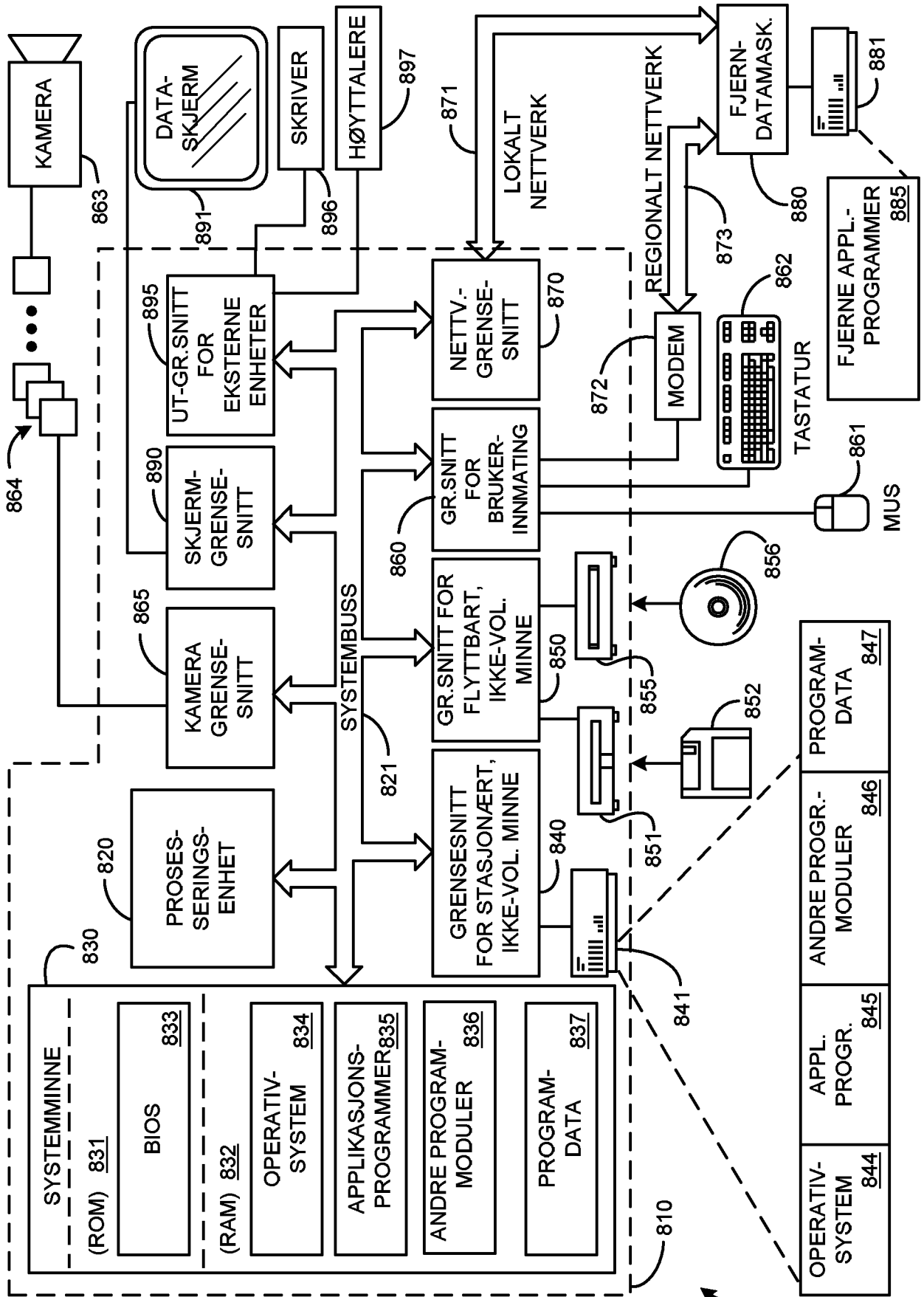


FIG. 8

800