(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 698 31 531 T2** 2006.05.18

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 886 238 B1**
(21) Deutsches Aktenzeichen: **698 31 531.6**
(96) Europäisches Aktenzeichen: **98 304 511.3**
(96) Europäischer Anmeldetag: **08.06.1998**
(97) Erstveröffentlichung durch das EPA: **23.12.1998**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **14.09.2005**
(47) Veröffentlichungstag im Patentblatt: **18.05.2006**

(51) Int Cl.[8]: ***G06K 15/02*** (2006.01)
G06T 1/00 (2006.01)

(30) Unionspriorität:
878978 **19.06.1997** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Xerox Corp., Rochester, N.Y., US

(74) Vertreter:
Grünecker, Kinkeldey, Stockmair & Schwanhäusser, 80538 München

(72) Erfinder:
Linder, Stephen F., Fairport, New York 14450, US;
Mallery, Craig H., Webster, US

(54) Bezeichnung: **Verfahren und Vorrichtung zur Verarbeitung und Wiedergabe von objektorientierten, neutralen Bilddaten**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die vorliegende Erfindung betrifft die Verarbeitung und Wiedergabe von objektorientierten Bilddaten in einem digitalen Farbdruck oder -Wiedergabesystem. Insbesondere betrifft die vorliegende Erfindung die Verarbeitung und Wiedergabe von objektorientierten neutralen Bilddaten getrennt von den nicht neutralen Bilddaten basierend auf der Objekttyp- und Objekt-Untertyp-Klassifizierung der Bilddaten.

[0002] Der Gebrauch von Farbe in dem digitalen Umfeld hat Probleme für Farbdrucker geschaffen, die zufriedenstellende Ergebnisse zu erzeugen versuchen. Ein Problem, dem Farbdrucker gegenüberstehen, stammt von der Verbreitung von Desktop-Publishing-Softwareprogrammen oder -Anwendungen.

[0003] Das Problem bei diesen Desktop-Publishingsystemen ist, dass die Systeme dem Benutzer erlauben, verschiedene Objekttypen in einem zusammengesetzten Dokument zu kombinieren. Ein Benutzer kann z.B. fotografische Bilder und Geschäftsgrafik (Charts) in einem einzigen Dokument kombinieren, wobei diese Bilder entweder in Farbe oder in Schwarz/Weiß sein können.

[0004] Um zufriedenstellende Ergebnisse zu erzielen, muss jedes dieser Objekte unterschiedlich verarbeitet werden, sodass das Dokument mit hoher Qualität erzeugt werden kann. Zum Beispiel sei angenommen, dass ein digitales Farbsystem versucht, ein Verbunddokument mit einem fotografischen Bild und einer Geschäftsgrafik wiederzugeben. Um eine hochwertige Wiedergabe eines fotografischen Bildes zu erzielen, kann es sein, dass das Farbsystem die Farbattribute in einer bestimmten Weise verschieben muss, aber dieses Verschieben kann bewirken, dass die Geschäftsgrafik in dem gleichen Verbunddokument ausgewaschen aussieht. Wenn andererseits das Farbdrucksystem verschoben wird, um satte, lebendige Farben für die Geschäftsgrafik zu gewährleisten, kann das fotografische Bild in dem Verbunddokument sein naturgetreues Aussehen verlieren.

[0005] Um dieses Problem zu lösen, sind objektorientierte Wiedergabesysteme entwickelt worden. Bei solchen herkömmlichen Systemen werden die Objekte, die ein Verbunddokument ausmachen, unterschiedlich wiedergegeben (verarbeitet). Mit anderen Worten, ein fotografisches Objekt oder Bild wird in einer Weise verarbeitet, während eine Geschäftsgrafik in einer anderen Weise verarbeitet wird. Auf diese Weise kann ein Objekt wiedergegeben werden, um seine Qualität zu optimieren. Beispiele eines solchen Systems werden in EP-A-0674289 Des Weiteren können Anforderungen des neutralen Wiedergebens für verschiedene Objekt typen variieren. Das heißt, neutrales Wiedergeben bezieht sich auf das Aussehen von Schwarz, Grau und Weiß auf einer gedruckten Seite oder Ausgabevorrichtung. Dieses Wiedergeben kann in Form von Prozess- (mehrfarbigen) oder echten (einzelne oder keine Farbe) Neutralen verstanden werden. Tabelle 1 unten veranschaulicht, wie neutrales Wiedergeben von Objekt zu Objekt variieren kann.

Tabelle 1

OBJEKT	SCHWARZ	GRAU	WEISS
Bitmap	Prozessschwarz	Prozessschwarz	muss nicht Papierweiß sein
Grafik	Einkomponenten-Schwarz	Einkomponenten-Schwarz	Papierweiß
Text	Einkomponenten-Schwarz	Einkomponenten-Schwarz	Papierweiß
Fotografisch	Prozessschwarz	Prozessschwarz	muss nicht Papierweiß sein

[0006] Wie in Tabelle 1 veranschaulicht, können Prozessschwarz und Grau für fotografische Objekte annehmbar sein, da echtes Schwarz oder Grau unerwünschte Glanzunterschiede einbringen kann. Andererseits wird für schwarze Linien in einem grafischen Objekt echtes Schwarz oder Grau gewünscht, um Fehlpassung und Artefakte von unvollkommener Farbbalance zu verringern. Ein Objekt kann daher erfordern, dass sein neutrales Bild in einer von dem Wiedergeben des nicht-neutralen Bildes abweichenden Weise wiedergegeben wird. Diese verschiedenen Forderungen verursachen Probleme, wenn Objekte in einem zusammengesetzten Bild wiedergegeben werden.

[0007] Ein Beispiel dieses Problems ist das Vorhandensein eines nennenswerten Glanzunterschieds, wenn schwarzer Text in einem Geschäftsgrafikobjekt wiederzugeben ist. Eine herkömmliche Lösung ist, das Objekt

unter Verwendung aller Primärfarben wiederzugeben, was zur Folge hat, dass der schwarze Text mit Prozessschwarz (Zyan, Magenta und Gelb) wiedergegeben wird. Durch Wiedergeben des Textes mit Prozessschwarz wird der Text hoch glänzend aussehen, was aus der Sicht eines Benutzers nicht unbedingt erwünscht ist. Die meisten Benutzer bevorzugen einen niedrigen Glanz, der mit Prozessschwarz nicht verwirklicht werden kann. Eine andere mögliche, herkömmliche Lösung ist von dem Standpunkt des Erlangens von Neutralen die Benuzung einer 100% Unterfarben-Entfernungs-(UCR) Strategie. Diese Strategie kann jedoch die Farbsättigung von Schattenbereichen reduzieren und unter anderen Farbtreueproblemen leiden.

[0008] Ein anderes Beispiel dieses Problems sind Weißpunktverschiebungen, die durch sachkundige Farbsteuerjustierungen verursacht werden, die dem Benutzer das Steuern verschiedener Farbattribute, wie Helligkeit, Kontrast, Farbsättigung und Farbbalance, erlauben. Weißpunktverschiebungen sind für einige Objekte wünschenswert, da diese Arten von Verschiebungen in natürlichen Szenen vorkommen; in Grafik- und Textobjekten ist es jedoch erwünscht, die fachkundige Farbsteuerjustierung die Weißpunkte unverändert und unverschoben zu lassen, da Weißpunktverschiebungen in Grafik und Textobjekten unerwünschte Artefakte erzeugen.

[0009] Nicht nur benötigen verschiedene Objekte unterschiedliche neutrale Wiedergabeprozesse, sondern die neutralen Elemente in einem Objekt können auch jeweils einmalige Wiedergabeprozesse erfordern. Zum Beispiel kann ein Objekt als ein grafisches Objekt eingestuft werden. Außerdem kann dieser grafische Objekttyp eine schwarze Linie und/oder Füllung enthalten. Obwohl sich schwarze Linien und Füllungen viele gemeinsame Wedergabe-Charakteristika als Mitglieder des grafischen Objekts teilen, unterscheiden sich schwarze Linien und Füllungen in Bezug auf neutrales Wiedergeben. Das heißt, schwarze Linien sollten als Einkomponenten-Schwarz und nicht als Prozessschwarz wiedergegeben werden, um Fehlpassungs- und unvollkommene Farbbalance-Artefakte zu verringern. Andererseits sollten schwarze Füllungen als Porzessschwarz und nicht als Einkomponenten-Schwarz wiedergegeben werden, um differenzielle Glanzeffekte zu reduzieren, die in schwarzen und weißen Abtastungen offensichtlich sein können, und um schwarze Füllungsausfälle infolge von Glanzunterschieden mit gefüllten Farbbereichen zu verringern.

[0010] Es ist daher wünschenswert, ein objektorientiertes Verarbeitungs- und Wiedergabesystem zu benutzen, das objektorientiertes neutrales Wiedergeben erlaubt, und ein Rahmenwerk für Merkmale bereitzustellen, die neutrales Wiedergeben durchführen. Außerdem ist es wünschenswert, ein objektorientiertes, neutrales Verarbeitungs- und Wiedergabesystem zu haben, das das Wiedergeben neutraler Farben basierend auf Unterteilen von Objekten und von Typen optimiert, die durch Unterteilen jedes Objekts in Objekt-Untertypen weiter optimiert werden können. Mit anderen Worten, es ist wünschenswert, ein Verarbeitungs- und Wiedergabesystem zu haben, das es erlaubt, gemeinsame Wiedergabemerkmale durch den Objekttyp zu identifizieren und eine Wiedergabe-Feinabstimmung basierend auf dem Objekt-Untertyp durchzuführen.

[0011] Eine Aspekt der vorliegenden Erfindung ist ein System zum Verarbeiten objektorientierter Bilddaten. Dieses System enthält eine erste Parser-Schaltung, um die objektorientierten Bilddaten in nicht-neutrale Farbbilddaten und neutrale Farbbilddaten zu zerlegen, eine zweite Parser-Schaltung, um die neutralen Farbbilddaten in Schwarzfarben-Bilddaten, Graufarben-Bilddaten und Weißfarben-Bilddaten zu zerlegen, und eine Neutralfarben-Verarbeitungsschaltung, um die Schwarzfarben-Bilddaten, Graufarben-Bilddaten und die Weißfarben-Bilddaten zu verarbeiten.

[0012] Ein anderer Aspekt der vorliegenden Erfindung ist ein Verfahren zum Verarbeiten objektorientierter Bilddaten. Dieses Verfahren zerlegt objektorientierte Bilddaten in nicht-neutrale Farbbilddaten und neutrale Farbbilddaten, zerlegt die neutralen Farbbilddaten in Schwarzfarben-Bilddaten, Graufarben-Bilddaten und Weißfarben-Bilddaten, verarbeitet die Schwarzfarben-Bilddaten, Graufarben-Bilddaten und die Weißfarben-Bilddaten und verarbeitet die Prozessschwarzfarben-Bilddaten, Prozessgraufarben-Bilddaten, die Prozessweißfarben-Bild-Daten und die Nicht-Neutralfarben-Bilddaten.

[0013] Das System kann unter Verwendung eines geeignet programmierten Computers oder einzelner Schaltungen implemeniert werden.

[0014] Weitere Aufgaben und Vorteile der vorliegenden Erfindung werden aus den folgenden Beschreibungen der verschiedenen Ausführungen und charakteristischen Merkmale der vorliegenden Erfindung ersichtlich werden.

[0015] Das Folgende ist eine Kurzbeschreibung jeder zum Beschreiben der vorliegenden Erfindung benutzten Zeichnung und wird daher nur zu Zwecken der Veranschaulichung vorgelegt und soll den Umfang der vor-

liegenden Erfindung nicht begrenzen. Inhalt der Zeichnungen:

[0016] Fig. 1 veranschaulicht ein Blockdiagramm eines typischen xerograhischen Farbwiedergabesystems.

[0017] Fig. 2 zeigt ein Blockschaltbild, das ein System zum Wiedergeben objektorientierter Bilddaten nach den Konzepten der vorliegenden Erfindung veranschaulicht.

[0018] Fig. 3 ist ein Flussdiagramm, das einen Prozess zum Wiedergeben objektorientierter Bilddaten nach den Konzepten der vorliegenden Erfindung veranschaulicht.

[0019] Fig. 4 zeigt ein Blockschaltbild, das ein System zum Verarbeiten objektorientierter Bilddaten nach den Konzepten der vorliegenden Erfindung veranschaulicht.

[0020] Fig. 5 zeigt Flussdiagramm, das die Verarbeitung objektorientierter Bilddaten nach den Konzepten der vorliegenden Erfindung veranschaulicht.

[0021] Fig. 6 zeigt ein Flussdiagramm, das das Wiedergeben objektorientierter Bilddaten nach den Konzepten der vorliegenden Erfindung veranschaulicht.

[0022] Fig. 7 zeigt ein Blockschaltbild, das ein System zum Verarbeiten objektorientierter Bilddaten nach den Konzepten der vorliegenden Erfindung veranschaulicht.

[0023] Fig. 8 ist ein Blockschaltbild, das ein herkömmliches objektorientiertes Wiedergabesystem zeigt.

[0024] Das Folgende wird eine ausführliche Beschreibung der in der vorliegenden Erfindung gezeigten Zeichnungen sein. in dieser Beschreibung sowie in den Zeichnungen stellen gleiche Verweiszeichen gleiche Vorrichtungen, Schaltungen oder Ersatzschaltungen dar, die die gleichwertigen Funktionen ausführen.

[0025] Wie oben erwähnt, ist die vorliegende Erfindung auf objektorientierter Drucken oder Wiedergeben gerichtet. Der wirkliche Wiedergabeprozess kann durch eine Vielfalt von Wiedergabeprozessen, wie z.B. Laser-Xerographie, Tintenstrahl usw., durchgeführt werden. Der Kürze halber wird ein laserxerographisches System unten in Bezug auf die vorliegende Erfindung beschrieben.

[0026] Bei dem laserxerographischen Druckprozess wird ein elektrostatisches Ladungsmuster oder latentes Bild, das einem Orginal entspricht, auf einem isolierenden Medium gebildet. Eine betrachtbare Aufzeichnung wird dann durch Entwickeln des latenten Bildes mit Partikeln aus granuliertem Material erzeugt, um ein Pulverbild davon zu bilden. Danach wird das sichtbare Pulverbild auf das isolierende Medium aufgeschmolzen oder auf ein geeignetes Trägermaterial übertragen und darauf aufgeschmolzen. Das Entwickeln des latenten Bildes er folgt, indem eine Entwicklermischung damit in Kontakt gebracht wird. Typische Entwicklermischungen umfassen im Allgemeinen gefärbte oder farbige thermoplastische Partikel aus granuliertem Material, in der Technik als Tonerpartikel bekannt, die mit Trägerkörnchen, z.B. ferromagnetische Körnchen, gemischt werden. Wenn angebracht, werden die Tonerpartikel mit Trägerkörnchen gemischt, und die Tonerpartikel werden triboelektrisch auf die richtige Polarität aufgeladen. Wenn die Entwicklermischung mit dem elektrostatischen latenten Bild in Kontakt gebracht wird, haften die Tonerpartikel daran. Da jedoch Tonerpartikel aus der Entwicklermischnung abgereichert werden, müssen zusätzliche Tonerpartikel zugeführt werden.

[0027] Fig. 1 zeigt ein Beispiel eines elektrostatischen Digital-Farbkopierers. Fig. 1 ist eine Draufsicht, die eine exemplarische elektrofotografische Druck-/Kopier-Maschine und einen vernetzten PC zeigt. Ein ESS (elektronisches Untersystem) oder Bildverarbeitungsstation (beide als IPS bezeichnet) allgemein durch Verweiszeichen **12** bezeichnet, enthält Datenverarbeitungs- und Steuerelektronik, die den Bilddatenfluss zu einem allgemein mit **16** bezeichneten Rasterausgabe-Scanner (ROS) herstellt und verwaltet. Ein Netzwerk aus einem oder mehreren Personal Computern (PC), allgemein mit **5** bezeichnet, wird als mit der IPS **12** ver bunden oder damit kommunizierend gezeigt. Eine Benutzerschnittstelle (UI), allgemein mit **14** bezeichnet, steht ebenfalls in Kommunikation mit der IPS **12**.

[0028] Die UI **14** ermöglicht einem Operator, verschiedene, operatoreinstellbare Funktionen und Wartungsaktivitäten zu steuern und zu überwachen. Der Operator betätigt geeignete Tasten der UI **14**, um die Parameter der Kopie einzustellen. Die UI **14** kann ein Berührungsschirm oder jedes andere geeignete Steuerfeld sein, das eine Operatorschnittstelle mit dem System bereitstellt. Das Ausgangssignal der UI **14** wird an die IPS **12** gesendet. Die UI **14** kann auch elektronische Dokumente auf einem (nicht gezeigten) Anzeigeschirm anzeigen

und die Bildwiedergabeauswahlen durchführen.

[0029] Wie weiter in Fig. 1 gezeigt, kann ein mehrfarbiges Originaldokument **38** auf einem (optionalen) mit **10** bezeichneten Rastereingabe-Scanner (RIS) positioniert werden. Der RIS enthält Dokument-Beleuchtungslampen, Optik, einen mechanischen Abtastantrieb und eine ladungsgekoppelte Vorrichtung (CCD-Anordnung) oder vollbreite Farbabtastanordnung. Der RIS **10** erfasst das ganze Bild von dem Originaldokument und wandelt es in eine Serie von Rasterabtastzeilen um und misst außerdem einen Satz von Primärfarbdichten, d.h. Dichten von Rot, Grün und Blau, an jedem Punkt des Originaldokuments. Der RIS **10** kann Daten des abgetasteten Bildes der IPS **12**, indirekt dem PC **5** und/oder direkt dem PC **5** zur Verfügung stellen.

[0030] Dokumente in digitaler oder anderer Form können erzeugt, gesiebt, modifiziert, gespeichert und/oder anderweitig durch den PC **5** verarbeitet werden, bevor sie zu der IPS **12** zum Drucken auf einem Drucker **18** gesendet/weitergeleitet werden. Das Display des PC **5** kann elektronische Dokumente auf einem Schirm (nicht gezeigt) anzeigen. Die IPS **12** kann den oder die Prozessor(en) und Controller (nicht gezeigt) enthaften, die zum Bewerkstelligen des adaptiven Bildwiedergabesystems der vorliegenden Erfindung nötig sind.

[0031] Die IPS **12** kann auch Signale, die dem gewünschten elektronischen oder abgetasteten Bild entsprechen, an den ROS **16** senden, der das Ausgabekopiebild erzeugt. Der ROS **16** enthält einen Laser mit rotierenden Polygonspiegelblöcken. Der ROS beleuchtet über den Spiegel **37** den geladenen Teil eines fotoleitfähigen Bandes **20** eines Druckers oder Markiergerätes, allgemein mit **18** bezeichnet, mit einer vorbestimmten Rate von MXN Pixeln pro Zoll, um einen Satz von subtraktiven, latenten Primärbildern zu erlangen. MXN kann 400 × 400 dpi (Punkte pro Zoll) oder auch asymmetrische Auflösungen wie etwa 300 × 1200 dpi darstellen.

[0032] Der ROS wird das fotoleitfähige Band belichten, um die latenten Bilder aufzuzeichnen, die den von der IPS **12** gesendeten Signalen entsprechen. Ein latentes Bild wird mit Zyan-Entwicklermaterial entwickelt. Ein anderes latentes Bild wird mit Magenta-Entwicklermaterial entwickelt, das dritte latente Bild wird mit Gelb-Entwicklermaterial entwickelt. Ein schwarzes latentes Bild kann anstelle von oder zusätzlich zu anderen (farbigen) latenten Bildern entwickelt werden. Diese entwickelten Bilder werden auf ein Kopierblatt in überlagerter Passung miteinender übertragen, um ein mehrfarbiges Bild auf dem Kopierblatt zu bilden. Dieses mehrfarbige Bild wird auf das Papierblatt aufgeschmolzen, um eine Farbkopie zu bilden.

[0033] Weiter auf Fig. 1 verweisend ist der Drucker oder Markiergerät **18** eine elektrofotografische Druckmaschine. Das fotoleitfähige Band **20** des Markiergerätes **18** besteht vorzugsweise aus einem fotoleitfähigen Material. Das fotoleitfähige Band bewegt sich in der Richtung von Pfeil **22**, um aufeinanderfolgende Abschnitte der fotoleitfähigen Oberfläche nacheinander durch die verschiedenen um den Bewegungsweg derselben angeordneten Verarbeitungsstationen zu transportieren. Das fotoleitfähige Band **20** wird um Walzen **23** und **26**, eine Spannwalze **28** und eine Antriebswalze **30** geführt. Die Antriebswalze **30** wird von einem Motor **32** gedreht, der durch eine geeignete Einrichtung, z.B. einen Riemenantrieb, damit verbunden ist. Wenn die Walze **30** sich dreht, transportiert sie das Band **20** in der Richtung von Pfeil **32**.

[0034] Zu Anfang durchläuft ein Teil des fotoleitfähigen Bandes **20** eine Ladestation, allgemein mit **33** bezeichnet. In der Ladestation **33** lädt eine Korona-Erzeugungsvorrichtung **34** das fotoleitfähige Band **20** auf ein relativ hohes, im Wesentlichen gleichförmiges Potential auf.

[0035] Als Nächstes wird die aufgeladene fotoleitfähige Oberfläche zu einer Belichtungsstation, all-gemein mit **35** bezeichnet, gedreht. Die Belichtungsstation **35** empfängt einen modulierten Lichtstrahl, der Information entspricht, die durch den RIS **10**, auf dem das mehrfarbige Originaldokument **38** positionert ist, gewonnen wird. Der modulierte Lichtstrahl trifft auf die Oberfläche des fotoleitfähigen Bandes **20**. Der Strahl beleuchtet den geladenen Teil des fotoleitfähigen Bandes, um ein elektrostatisches latentes Bild zu formen. Das fotoleitfähige Band wird drei oder vier Mal belichtet, um drei oder vier latente Bilder darauf auzuzeichnen. Wenn ein Hifi-Farbsystem benutzt wird, wird das fotoleitfähige Band für jede Primärfarbe in dem Entwicklungssystem belichtet werden. Das heißt, in einem Hifi-Farbsystem mit den Primärfarben Zyan, Magenta, Gelb, Orange und Schwarz wird das fotoleitfähige Band **5** Mal belichtet werden.

[0036] Nachdem die elektrostatischen latenten Bilder auf dem fotoleitfähigen Band **20** aufgezeichnet sind, transportiert das Band diese latenten Bilder zu einer Entwicklungsstation, allgemein mit **39** bezeichnet. Die Entwicklungsstation enthält vier einzelne Entwicklereinheiten, die mit **40**, **42**, **44** und **46** bezeichnet sind. Die Entwicklereinheiten sind von einem Typ, der gewöhnlich in der Technik als "Magnetbürsten-Entwicklungseinheit" bezeichnet wird.

[0037] Ein Magnetbürsten-Entwicklungssystem verwendet typischerweise ein magnetisierbares Entwicklermatertal, das magnetische Trägerkörnchen mit Tonerpartikeln enthält, die triboelektrisch daran haften. Das Entwicklermaterial wird fortlaufend durch ein gerichtetes Magnetflussfeld gebracht, um eine Bürste aus Entwicklermaterial zu bilden. Das Entwicklermaterial ist ständig in Bewegung, sodass die Bürste laufend mit frischem Entwicklermaterial versorgt wird. Die Entwicklung erfolgt, indem die Bürste aus Entwicklermatertal mit der fotoleitfähigen Oberfläche in Kontakt gebracht wird. Die Entwicklereinheiten **40**, **42** und **44** verwenden Tonerpartikel einer spezifischen Farbe, die dem Komplement des spezifischen farbseparierten elektrostatischen latenten Bildes entspricht, das auf der fotoleitfähigen Oberfläche aufgezeichnet ist.

[0038] Die Farbe jedes der Tonerpartikel absorbiert Licht in einem vorbestimmten Spektralbereich des elektromagnetischen Wellenspektrums. Zum Beispiel wird ein durch Entladen der Ladungsteile auf den fotoleitfähigen Band, die den grünen Bereichen des Originaldokuments entsprechen, gebildetes elektrostatisches latentes Bild die roten und blauen Abschnitte als Bereiche von relativ hoher Ladungsdichte auf dem fotoleitfähigen Band **20** aufzeichnen, während die grünen Bereiche auf einen für die Entwicklung unwirksamen Spannungspegel reduziert werden. Die geladenen Bereiche werden dann sichtbar gemacht, indem die Entwicklereinheit **40** grünabsorbierende (Magenta) Tonerpartikel auf das auf dem fotoleitfähigen Band **20** aufgezeichnete elektrostatische latente Bild aufbringt.

[0039] Desgleichen wird eine Blauseparation durch die Entwicklereinheit **42** mit blauabsorbierenden (Gelb) Tonerpartikeln entwickelt, während die Rotseparation durch die Entwicklereinheit **44** mit rotabsorbierenden (Zyan) Tonerpartikeln entwickelt wird. Die Entwicklereinheit **46** enthält schwarze Tonerpartikel und kann benutzt werden, um das aus einem Schwarz/Weiß-Originaldokument gebildete elektrostatische latente Bild zu entwickeln. Jede der Entwicklereinheiten wird zu einer Betriebsposition bewegt bzw. davon wegbewegt. In der Betriebsposition ist die Magnetbürste im Wesentlichen angrenzend an das fotoleitfähige Band **20**, während in der Nicht-Betriebsposition die Magnetbürste davon beabstanded ist. Während des Entwickelns jedes elektrostatischen latenten Bildes befindet sich nur eine Entwicklereinheit in der Betriebsposition, die übrigen Entwicklereinheiten befinden sich in der Nicht-Betriebsposition.

[0040] Nach dem Entwickeln wird das Tonerbild zu einer allgemein mit **65** bezeichneten Transferstation bewegt. Die Transferstation **65** entält eine allgemein mit **64** bezeichnete Transferzone. In der Transferzone **64** wird das Tonerbild auf ein Blatt aus Trägermaterial, z.B. einfaches Papier, übertragen. In der Transferstation **65** bewegt ein allgemein mit **48** bezeichneter Blatttransportmechanismus das Blatt in Kontakt mit dem fotoleitfähigen Band **20**. Der Blatttransport **48** besitzt ein Paar beabstandeter Bänder **54**, die um ein Paar im Wesentlichen zylindrischer Walzen **50** und **53** geführt werden. Ein Blattgreifer (nicht gezeigt) liegt zwischen den Bändern **54** und bewegt sich unison damit.

[0041] Ein Blatt **25** wird von einem auf einer Schale angeordneten Blattstapel entnommen. Ein Friktionshemmungs-Zuführer **58** transportiert das oberste Blatt von dem Stapel **56** zu einem Vortransfer-Transport **60**. Der Transport **60** transportiert das Blatt (nicht gezeigt) zu einem Blatttransport **48**. Das Blatt wird durch den Transport **60** synchron mit der Bewegung des Blattgreifers transportiert. Der Blattgreifer schließ sich dann, um das Blatt zur Bewegung damit in einem umlaufenden Pfad festzuhalten. Die Vorderkante des Blattes (nicht gezeigt) wird so gehalten, dass es durch den Blattgreifer losgelassen werden kann.

[0042] Wenn sich die Bänder **54** in der Richtung von Pfeil **62** bewegen, bewegt sich das Blatt synchron mit dem darauf entwickelten Tonerbild in Kontakt mit dem fotoleitfähigen Band **20**. In der Transferzone **64** sprüht eine Koronaerzeugungsvorrichtung **66** Ionen auf die Rückseite des Blattes, um so das Blatt auf die richtige Größe und Polarität aufzuladen, um das Tonerbild von dem fotoleitfähigen Band **20** anzuziehen. Das Blatt bleibt fest an dem Blattgreifer, um sich für drei Zyklen in einem umlaufenden Pfad zu bewegen. Auf diese Weise werden drei oder vier Tonerbilder verschiedener Farbe auf das Blatt in überlagerter Passung miteinander übertragen.

[0043] Ein Fachmann wird erkennen, dass sich das Blatt in einem umlaufenden Pfad für vier Zyklen bewegen kann, wenn Unterfarben-Entfernung (UCR) benutzt wird. Jedes der auf der fotoleitfähigen Oberfläche aufgezeichneten elektrostatischen latenten Bilder wird mit geeignet gefärbten Toner entwickelt und in überlagerter Passung miteinander auf das Blatt übertragen, um die mehrfarbige Kopie des farbigen Originaldokuments zu bilden. Nach dem letzten Transfervorgang dirigiert das Blatttarnsportsystem das Blatt zu einer Vakuum-Transporteinrichtung **68**. Die Vakuum-Transporteinrichtung **68** transportiert das Blatt in der Richtung von Pfeil **70** zu einer allgemein mit **71** bezeichneten Aufschmelzstation, wo das übertragene Tonerbild dauerhaft auf das Blatt geschmolzen wird. Danach wird das Blatt durch ein Walzenpaar **76** zu einer Auffangschale **78** transportiert, um dann von dem Maschinenoperator entnommen zu werden.

[0044] Die letzte Verarbeitungsstation in der Bewegungsrichtung des Bandes **20**, wie durch Pfeil **22** angegeben, ist eine Photorezeptor-Reinigungsvorrichtung, allgemein mit **73** bezeichnet. Eine drehbar montierte Faserbürste **72** kann in der Reinigungsstation positioniert sein und mit dem fotoleitfähigen Band **20** in Kontakt gehalten werden, um restliche Tonerpartikel, die nach dem Transfervorgang übrigbleiben, zu entfernen. Danach beleuchtet eine Lampe **82** das fotoleitfähige Band **20**, um vor dem Beginn des nächsten folgenden Zylusses die restliche darauf verbliebene Ladung zu entfernen.

[0045] Wie oben erwähnt, geben herkömmliche objektorientiere Wiedergabesysteme basierend auf dem Objekt wieder, ungeachtet des Inhalts des Objekts, z.B., ob das Objekt neutrale Bilddaten enthält oder nicht. Ein Beispiel eines solchen Systems wird in Fig. 8 veranschaulicht, wo ein Trio von Daten (Farbe, Farbraum und Objektinformation) in eine Wiedergabe pfadschaltung **105** geführt wird, die das Trio entsprechend der Objektbezeichnung verarbeiten wird. Nach dem Verarbeiten werden die Daten zur Wedergabe zu einer Wiedergabevorrichtung **106** geleitet. Durch Benutzen dieses herkömmlichen Systems, um sowohl neutrale Bilddaten als auch nicht-neutrale Bilddaten zu verarbeiten und wiederzugeben, opfert das herkömmliche Wiedergabesystem Farbtreue.

[0046] Um diese Probleme bei den herkömmlichen Vorrichtungen zu vermeiden, transformiert die vorliegende Erfindung, wie in Fig. 2 gezeigt, die neutralen Farbdaten, bevor sie an die Wiegergabepfadschaltung **105** geschickt werden. Das heißt, objektorientierte Bilddaten werden durch eine Neutralwiedergabe-Transformationsschaltung **100** empfangen, die die ankommenden Bilddaten in neutrale Bilddaten und nicht-neutrale Bilddaten zerlegt. Die Farbbilddaten und Farbraumdaten der zerlegten neutralen Bilddaten werden transformiert, und die transformierten neutralen Bilddaten und die nicht-neutralen Bilddaten werden in ein Abbildungssystem **105** geführt.

[0047] Danach werden die verarbeiteten neutralen Farbbilddaten und die verarbeiteten nicht-neutralen Farbbilddaten in eine Wiedergabevorrichtung **106** geführt, die die Bilddaten in ein Bild umwandelt, das durch einen Benutzer leicht zu sehen ist. Diese Wiedergabevorrichtung kann entweder ein Anzeigeschirm oder ein Drucksystem sein.

[0048] Fig. 3 veranschaulicht den von dem in Fig. 2 gezeigten System benutzten Prozess. Das heißt, wie in Fig. 3 gezeigt, empfängt Schritt S1 Bilddaten in der Form von Bilddaten-Trios. Ein Bilddaten-Trio besteht aus Farbbilddaten, Daten, die den Farbraum angeben, und Daten, die den Objekttyp der ankommenden Bilddaten angeben. Der ankommende Strom von Farbbilddaten-Trios (Farbbilddaten, Farbraumdaten und Objekttypdaten) wird in Schritt S2 basierend auf dem mit dem Wiedergeben des Objekts verbundenen Merkmalsatz zerlegt. Diese ankommenden Trios werden in Schritt S2 in zwei Sätze von Trios zerlegt. Ein Satz von Trios enthält neutrale Farben, und der andere Satz von Trios enthält nicht-neutrale Farben.

[0049] Die Trios, die neutrale Bilddaten enthalten, lassen ihre Farb- und Farbrauminformation in Schritt S7 transformieren. Danach werden die transformierten neutralen Bilddaten und die nicht-neutralen Bilddaten unter Verwendung eines objektabhängigen Vorrichtungs-Seitenweges verarbeitet.

[0050] Bei einer Ausführung der vorliegenden Erfindung, wie in Fig. 4 veranschaulicht, werden die neutralen Farbbilddaten weiter in getrennte Komponenten zerlegt. Das heißt, wie in Fig. 4 gezeigt, eine Neutral-/Nicht-Neutral-Parser-Schaltung **100** empfängt einen ankommenden Strom von Trios, die Farbbilddaten, Farbraumdaten und Objekttypdaten enthalten. Diese Trios werden durch die Neutral-/Nicht-Neutral-Parser-Schaltung **100** in zwei Sätze von Trios zerlegt, wobei ein Satz von Trios neutrale Farben enthält, und der andere Satz von Trios nicht neutrale Farben enthält. Der Satz, der nicht-neutrale Farben enthält, wird in der üblichen, herkömmlichen Weise durch das Abbildungssystem **105** verarbeitet.

[0051] Andererseits werden die Trios, die die neutralen Farbbilddaten (neutrale Farbbilddaten, Farbraumdaten und Objekttypdaten) enthalten, in eine neutrale Parser-Schaltung **101** geführt, die die ankommenden Trios weiter in drei getrennte Sätze von Trios basierend auf dem Merkmalsatz zerlegt.

[0052] Ein Triosatz enthält schwarze Trios, ein zweiter Triosatz enthält graue Trios, und ein dritter Triosat enthält weiße Trios, wobei jedes Trio Farbbilddaten, Farbraumdaten und Objekttypdaten enthält. Die sich ergebenden zerlegten Trios werden dann entsprechend dem Merkmalsatz unabhängig verarbeitet, um ihre Farbbilddaten und Farbraumdaten zu transformieren.

[0053] Das heißt, die Trios, die den schwarzen Bilddaten entsprechen, werden durch einen Schwarzprozessor **102** verarbeitet, während die Trios mit den grauen Bilddaten durch einen Grauprozessor **103** verarbeitet

werden. Schließlich werden die Trios, die die weißen Bilddaten enthalten, durch einen Weißprozessor **104** verarbeitet. Der Weißprozessor **102**, der Grauprozessor **103** und der Weißprozessor **104** sind in einer neutralen Verarbeitungsschaltung **110** enthalten. Die verarbeiteten neutralen Daten werden dann in das Abbildungssystem **105** geführt, wo diese neutralen Bilddaten durch den Rest das Systems in der herkömmlichen Weise verarbeitet werden, um eine Ausgabe zu erzeugen, die die richtigen neutralen Wiedergabeanforderungen erfüllt.

[0054] Fig. 5 veranschaulicht ein Verfahren zum Wiedergeben von neutralen Farbbilddaten nach den Konzepten der vorliegenden Erfindung und dem in Fig. 4 veranschaulichten System. Wie in Fig. 5 gezeigt, empfängt Schritt S1 Bilddaten in der Form von Trios. Diese Bilddaten werden in Schritt S2 in neutrale Farbbilddaten und nicht-neutrale Farbbilddaten zerlegt. Die zerlegten neutralen Farbbilddaten werden dann in Schritt S3 in schwarze Bilddaten, graue Bilddaten und weiße Bilddaten zerlegt. Die schwarzen Bilddaten-, grauen Bilddaten- und weißen Bilddaten-Trios werden dann in Schritt S4 jeweils einzeln verarbeitet. Die Prozessschwarz-Bilddaten-, Graubilddaten- und Weißbilddaten-Trios werden dann in Schritt S5 wieder in einem Abbildungssystem zusammen mit den nicht-neutralen Bilddaten verarbietet.

[0055] Ein Beispiel einer Software-Implementierung des oben beschriebenen Verfahrens wird in Tabelle 2 unten dargelegt. Diese Tabelle legt ein Beispiel eines Softwarecodes dar, wie in PostScript™ und C implementiert, der die oben beschriebenen Funktionen durchführt. Obwohl der Prozess der vorliegenden Erfindung in Software implementiert wurde, kann er ohne weiteres auch in Hardware implementiert werden.

Tabelle 2

```
NOTICE:    Copyright 1996, 1997 Xerox Corporation
           All Rights Reserved
 *
 *         Contains proprietary information of
 *         Xerox Corporation and is protected under the copyright
 *         laws as an unpublished work.
 *
 *

PUBLIC procedure neutralRenderingInit() ;
PUBLIC procedure preNeutralRendering() ;
PUBLIC procedure graphicPostNeutralRendering() ;
PUBLIC procedure textPostNeutralRendering() ;
PUBLIC procedure preStringwidth() ;
PUBLIC procedure postStringwidth() ;

#include PACKAGE_SPECS
#include CUSTOMOPS
#include UTIL
#include "neutralRendering.h"

/*************************************/
#define RETURN_BOOL
```

```
#define PATTERN_PAINT_TYPE_1
#define TRUE 1
#define FALSE 0
#define CURRENTPOINT        40
#define MOVETO              107
#define CURRENTCOLOR        213
#define CURRENTCOLORSPACE   214
#define SETCOLOR            221
#define SETCOLORSPACE       222
#define CURRENTOVERPRINT    433
#define SETOVERPRINT        450
#define DEVICE_CIEBASEDABC  471
#define DEVICE_CIEBASEDDEFG 478  /* CIEBasedDEFG isn't
defined by Adobe */
#define DEVICE_CIEBASEDA    470
#define DEVICE_CMYK 472
#define DEVICE_GREY 473
#define DEVICE_RGB  474
#define PATTERN     476
#define GSAVE       78
#define GRESTORE    77
#define POP         117
#define SETGREY     150
#define RGB_BLACK 0
#define RGB_WHITE 765   /* 3*255 */
#define SET_BLACK_OVERPRINT  " gsave true setoverprint [
/Separation /Black /DeviceCMYK {pop 0 0 0 1} ] setcolorspace 1
setcolor"
#define SET_BLACK_OVERPRINT2 " {true setoverprint [ /Separation
/Black /DeviceCMYK {pop 0 0 0 1} ] setcolorspace 1 setcolor}"
#define SET_BLACK_OVERPRINT3 " {X_SetBlackOverPrint}"
#define SET_GREY            " {XSD_setgrey}"

/*************************************************************************
***********/
PRIVATE boolean  gs_modified;
PRIVATE Int32 neutralNestingCount;
#ifdef RETURN_BOOL
PRIVATE boolean  return_status;
```

```
#endif

/*****************************************************************************
***********/
PRIVATE procedure setBlackOverprint( ) {
  gs_modified = TRUE;
#ifndef  RETURN_BOOL
  PSExecuteString( SET_BLACK_OVERPRINT );
#else
  PSExecuteString( SET_BLACK_OVERPRINT2 );
  return_status = TRUE;
#endif
}

/*****************************************************************************
***********/
PRIVATE procedure setGrey( real greyValue ) {
  gs_modified = TRUE;
#ifndef  RETURN_BOOL
  PSExecuteOperator ( GSAVE );
  PSPushReal( greyValue );
  PSExecuteOperator ( SETGREY );
#else
  PSPushReal( greyValue );
  PSExecuteString( SET_GREY );
  return_status = TRUE;
#endif
}

/*****************************************************************************
***********/
PRIVATE procedure processRGB( ) {
 boolean trueGrey;
 boolean blackOverPrint;
 boolean processBlack;
 Int32  component1, component2, component3, colorTemp, colorDiff;
 real   realComponent1;
 realComponent1 = PSPopReal();
 component1 =    realComponent1 * 255;
```

```
    component2 =    PSPopReal() * 255;
    component3 =    PSPopReal() * 255;
    trueGrey = PSPopBoolean();
    blackOverPrint = PSPopBoolean();
    processBlack = PSPopBoolean();
    colorTemp = component1 + component2 + component3;
    if ( colorTemp ==  RGB_BLACK ) {
     if (blackOverPrint) {
      setBlackOverprint();
     } else if (trueGrey && !processBlack) {
      setGrey( 0.0 );
     }
    } else if ( (colorTemp == RGB_WHITE) && trueGrey ) {
     setGrey(1.0);
    } else if (trueGrey) {
#if 0
     colorDiff = component1 - component2;
     colorTemp = os_abs(colorDiff);
     colorDiff = component1 - component3;
     colorTemp += os_abs(colorDiff);
     colorDiff = component2 - component3;
     colorTemp += os_abs(colorDiff);
     if (colorTemp <= 3) {
#else
     if ( (component1 == component2) && (component1 ==
component3) ) {
#endif
      setGrey( realComponent1 );
     }
    }
}

/**************************************************************************
***********/
PRIVATE procedure processGrey( ) {
 boolean blackOverPrint;
 real  realComponent1;
 realComponent1 = PSPopReal();
 blackOverPrint = PSPopBoolean();
```

```
 PSExecuteOperator(POP);  /* processBlack */
 if ( blackOverPrint && (realComponent1 == 0.0) ) {
    setBlackOverprint();
 }

/**********************************************************************
***********/
PRIVATE procedure processCMYK( ) {
 boolean trueGrey;
 boolean blackOverPrint;
 boolean processBlack;
 Int32  yellow, magenta, cyan, black, colorTemp, colorDiff;
 black =   PSPopReal() * 255;
 yellow =  PSPopReal() * 255;
 magenta = PSPopReal() * 255;
 cyan =    PSPopReal() * 255;
 trueGrey = PSPopBoolean();
 blackOverPrint = PSPopBoolean();
 processBlack = PSPopBoolean();
#if 0
 colorDiff = yellow - magenta;
 colorTemp = os_abs(colorDiff);
 colorDiff = yellow - cyan;
 colorTemp += os_abs(colorDiff);
 colorDiff = magenta - cyan;
 colorTemp += os_abs(colorDiff);
 if ( colorTemp <= 3 ) {
#else
 if ( (yellow == magenta) && (yellow == cyan) ) {
#endif
   /* neutral */
   if ( blackOverPrint && (black==255) ){
      setBlackOverprint();
   } else if ( trueGrey ) {
     if ( (black==0) && (cyan==0) && (magenta==0) && (yellow==0) ) {
       /* white */
       setGrey( 1.0 );
     } else {
```

```
        colorTemp = 255 - (black + cyan);
        if (colorTemp > 0) {
          /* grey */
          setGrey((real)colorTemp/255);
        } else {
          /* black */
          if (!processBlack) setGrey ( 0.0 );
        }
      }
    }
  }

}

/**********************************************************************
***********/
PRIVATE procedure processCIEBasedABC( ) {
  PSObject rangeABC;
  PSPopTempObject( arrayType, &rangeABC );
  if ( (rangeABC.val.arrayval[0].val.ival == 0) &&
       (rangeABC.val.arrayval[1].val.ival == 1) &&
       (rangeABC.val.arrayval[2].val.ival == 0) &&
       (rangeABC.val.arrayval[3].val.ival == 1) &&
       (rangeABC.val.arrayval[4].val.ival == 0) &&
       (rangeABC.val.arrayval[5].val.ival == 1) ) {
    processRGB();
  } else {
    PSExecuteOperator(POP);  /* C */
    PSExecuteOperator(POP);  /* B */
    PSExecuteOperator(POP);  /* A */
    PSExecuteOperator(POP);  /* trueGrey */
    PSExecuteOperator(POP);  /* blackOverPrint */
    PSExecuteOperator(POP);  /* processBlack */
  }
}

/**********************************************************************
***********/
PRIVATE procedure processCIEBasedDEFG( ) {
```

```
  PSObject rangeDEFG;
  PSPopTempObject( arrayType, &rangeDEFG );
  if ( (rangeDEFG.val.arrayval[0].val.ival == 0) &&
       (rangeDEFG.val.arrayval[1].val.ival == 1) &&
       (rangeDEFG.val.arrayval[2].val.ival == 0) &&
       (rangeDEFG.val.arrayval[3].val.ival == 1) &&
       (rangeDEFG.val.arrayval[4].val.ival == 0) &&
       (rangeDEFG.val.arrayval[5].val.ival == 1) &&
       (rangeDEFG.val.arrayval[6].val.ival == 0) &&
       (rangeDEFG.val.arrayval[7].val.ival == 1) ) {
     processCMYK();
   } else {
     PSExecuteOperator(POP); /* G */
     PSExecuteOperator(POP); /* F */
     PSExecuteOperator(POP); /* E */
     PSExecuteOperator(POP); /* D */
     PSExecuteOperator(POP); /* trueGrey */
     PSExecuteOperator(POP); /* blackOverPrint */
     PSExecuteOperator(POP); /* processBlack */
   }
}

/*************************************************************************
************/
PRIVATE procedure processCIEBasedA( ) {
  PSObject rangeA;
  boolean  blackOverPrint;
  boolean  trueGrey;
  real     realComponent1;
  PSPopTempObject( arrayType, &rangeA );
  if ( (rangeA.val.arrayval[0].val.ival == 0) &&
       (rangeA.val.arrayval[1].val.ival == 1) ) {
     realComponent1 = PSPopReal();
     trueGrey = PSPopBoolean();
     blackOverPrint = PSPopBoolean();
     PSExecuteOperator(POP); /* processBlack */
   if ( blackOverPrint && (realComponent1 == 0.0) ) {
      setBlackOverprint();
    } else if (trueGrey) {
```

```
    setGrey (realComponent1);
   }
  } else {
   PSExecuteOperator(POP); /* A */
   PSExecuteOperator(POP); /* trueGrey */
   PSExecuteOperator(POP); /* blackOverPrint */
   PSExecuteOperator(POP); /* processBlack */


  }
}
/*****************************************************************
***********/
PRIVATE procedure processPATTERN( ) {
 integer PaintType;
 PaintType = PSPopInteger();
 PSExecuteOperator(POP); /* blackOverPrint */
 PSExecuteOperator(POP); /* processBlack */
#ifdef PATTERN_PAINT_TYPE_1
 if ((PaintType == 1) && (neutralNestingCount == 1)) {
  neutralNestingCount=0;
 }
#endif
}

/*****************************************************************
***********/
PUBLIC procedure neutralRenderingInit() {
 gs_modified = FALSE;
 neutralNestingCount = 0;
}

/*****************************************************************
***********/
PUBLIC procedure preNeutralRendering() {
 Int32   colorspaceID;
 colorspaceID = PSPopInteger();
#ifdef RETURN_BOOL
 return_status = FALSE;
```

```
#endif
  neutralNestingCount++;
  if (neutralNestingCount > 1) {
    /* remove parameters from the stack */
    switch (colorspaceID) {
    case DEVICE_RGB:
      PSExecuteOperator(POP); /* blue */
      PSExecuteOperator(POP); /* green */
      PSExecuteOperator(POP); /* red */
      PSExecuteOperator(POP); /* trueGrey */
      break;
    case DEVICE_CMYK:
      PSExecuteOperator(POP); /* black */
      PSExecuteOperator(POP); /* yellow */
      PSExecuteOperator(POP); /* magenta */
      PSExecuteOperator(POP); /* cyan */
      PSExecuteOperator(POP); /* trueGrey */
      break;
    case DEVICE_GREY:
      PSExecuteOperator(POP); /* grey */
                        /* trueGrey not sent */
      break;
    case DEVICE_CIEBASEDABC:
      PSE_ ;cuteOperator(POP); /* range array */
      PSExecuteOperator(POP); /* C */
      PSExecuteOperator(POP); /* B */
      PSExecuteOperator(POP); /* A */
      PSExecuteOperator(POP); /* trueGrey */
      break;
    case DEVICE_CIEBASEDDEFG:
      PSExecuteOperator(POP); /* range array */
      PSExecuteOperator(POP); /* G */
      PSExecuteOperator(POP); /* F */
      PSExecuteOperator(POP); /* E */
      PSExecuteOperator(POP); /* D */
      PSExecuteOperator(POP); /* trueGrey */
      break;
    case DEVICE_CIEBASEDA:
      PSExecuteOperator(POP); /* range array */
```

```
    PSExecuteOperator(POP); /* A */
    PSExecuteOperator(POP); /* trueGrey */
    break;
  case PATTERN:
    PSExecuteOperator(POP); /* PaintType */
                    /* trueGrey not sent */
    break;
  }

  PSExecuteOperator(POP);   /* processBlack */
  PSExecuteOperator(POP);   /* blackOverPrint */
} else {
  switch (colorspaceID) {
  case DEVICE_RGB:
    processRGB();
    break;
  case DEVICE_GREY:
    processGrey();
    break;
  case DEVICE_CMYK:
    processCMYK();
    break;
  case DEVICE_CIEBASEDABC:
    processCIEBasedABC();
    break;
  case DEVICE_CIEBASEDDEFG:
    processCIEBasedDEFG();
    break;
  case DEVICE_CIEBASEDA:
    processCIEBasedA();
    break;
  case PATTERN:
    processPATTERN();
    break;
  default:
    PSExecuteOperator(POP);   /* processBlack */
    PSExecuteOperator(POP);   /* BlackOverprint */
    break;
  }
```

```
 }
#ifdef  RETURN_BOOL
 PSPushBoolean(return_status);
#endif
}

/*****************************************************************************
************/
PUBLIC procedure graphicPostNeutralRendering() {
#ifdef RETURN_BOOL
 boolean status = FALSE;
#endif
 neutralNestingCount --;
 if ( (neutralNestingCount == 0) && gs_modified ) {
   gs_modified = FALSE;
#ifndef  RETURN_BOOL
   PSExecuteOperator( GRESTORE );
#else
   status = TRUE;
#endif
 }
#ifdef PATTERN_PAINT_TYPE_1
 else if (neutralNestingCount < 0) {
   neutralNestingCount=0;
 }
#endif
#ifdef RETURN_BOOL
 PSPushBoolean(status);
#endif
}

/*****************************************************************************
************/
PUBLIC procedure textPostNeutralRendering() {
#ifdef RETURN_BOOL
 boolean status = FALSE;
#endif
  neutralNestingCount --;
  if ( (neutralNestingCount == 0) && (gs_modified ) ){
```

```
    gs_modified = FALSE;
#ifndef  RETURN_BOOL
    PSExecuteOperator( CURRENTPOINT );
    PSExecuteOperator( GRESTORE );
    PSExecuteOperator( MOVETO );
#else
    status = TRUE;
#endif

  }
#ifdef PATTERN_PAINT_TYPE_1
  else if (neutralNestingCount < 0) {
    neutralNestingCount=0;
  }
#endif
#ifdef RETURN_BOOL
  PSPushBoolean(status);
#endif
}

/*****************************************************************************
***********/
PUBLIC procedure preStringwidth() {
  neutralNestingCount++;
}

/*****************************************************************************
***********/
PUBLIC procedure postStringwidth() {
  neutralNestingCount--;
}
```

[0056] Das oben beschriebene neutrale Wiedergabesystem und -Verfahren vermeidet die mit Systemen verbundenen Probleme, die fachkundige Farbeinstellungen vornehmen, um verschiedene Farbmerkmale wie Helligkeit, Kontrast, Farbsättigung, Farbbalance usw. einzustellen. Zum Beispiel kann ein Experten-Steuersystem so programmiert werden, dass die Neutralen für bestimmte Objekttypen, z.B. Text und Grafik, durch die Experten-Steuerung unverändert bleiben sollen. Das neutrale Wiedergabesystem der vorliegenden Erfindung zerlegt und verarbeitet die neutralen Farb, Farbraum- und Objekttyp-Trios in einem Pfad des Systems, der von den Experten-Steuerungen nicht beeinflusst wird.

[0057] Außerdem stellt die vorliegende Erfindung weitere Vorteile bereit, indem sie ein objektorientiertes Wiedergabesystem benutzt, das Objekte in Objekttypen und Objekt-Untertypen einteilt. Eine ausführlichere Erklärung dieses Systems wird unten in der Beschreibung von Fig. 6 und Fig. 7 dargelegt.

[0058] Wie oben erörtert, optimieren objektorientierte Wiedergabesysteme das Wiedergeben basierend auf Unterteilung von Objekten in Typen, wie z.B. Bitmap, Text und Grafik. Dieses Markierungsverfahren hat bestimmte Grenzen, die durch Untertypisierung überwunden werden können. Das Objekttyp-/Objekt-Untertyp-Markierungsverfahren erlaubt es, gemeinsame Wiedergabemerkmale durch den Objekttyp zu identifizieren, und Wiedergabe-Feinabstimmung basierend auf dem Untertyp durchzuführen.

[0059] Als ein Beispiel können Sweeps oder Gradientenfüller als ein Untertyp eines Grafikobjekttyps in Betracht gezogen werden. Bestimmte Wiedergabemerkmale, wie Farbwiedergabe, können mit dem Grafikobjekttyp geteilt werden, während andere Merkmale wie Halbtonung für Sweeps optimiert werden können. Bei einer bevorzugten Ausführung der vorliegenden Erfindung sind die Objekttyp-/Untertyp-Kombinationen wie folgt:

Objekttyp	Objekt-Untertyp	Beispiel
Bitmap	Bild	Bild einer wirklichen Szene
Bitmap	Grafik	abgetastetes Firmenlogo
Grafik	Füller	Grafikbereichsfüller
Grafik	Linie	
Grafik	Sweep	Gradientenfüller
Text	halbgetont	
Text	kräftig	

[0060] Um dieses Konzept weiter zu erklären wird angenommen, dass es drei Objekttypen (Bitmap, Grafik und Text) und zwei Untertypen für grafische Objekte (Linien und Füller) gibt. Linien- und Füller-Untertypen teilen sich viele gemeinsame Wiedergabemerkmale als Mitglieder des Grafikobjekttyps, unterscheiden sich aber in Bezug auf neutrales Wiedergeben. Schwarze Linien werden als einzelne Komponente und nicht als Prozess wiedergegeben, um Fehlpassungs- und unvollkommene Farbbalance-Artefakte zu reduzieren. Schwarze Füller werden als Prozessschwarz und nicht als Einkomponenten-Schwarz wiedergegeben, um Differerenz-Glanzeffekte zu reduzieren, die in schwarzen und weißen Sweeps offensichtlich sind, und um schwarze Füller-"Ausfälle" infolge von Glanzunterschieden mit gefüllten farbigen Bereichen.

[0061] Fig. 6 veranschaulicht einen Prozess zum Wiedergeben neutraler Farbbilddaten basierend auf den oben erörterten Klassifizierungskonzepten von Objekttypen und Objekt-Untertypen. Wie in Fig. 6 gezeigt, bestimmt der Prozess in Schritt S10 einen Objekttyp für die ankommenden objektorientierten Bilddaten. Nach Bestimmen des Objekttyps wird in Schritt S11 ein Merkmalsatz basierend auf dem bestimmten Objekttyp erzeugt. Die objektorientierten Bilddaten werden weiter klassifiziert, um in Schritt S12 einen Objekt-Untertyp zu bestimmen, wobei die Untertyp-Bestimmung des Objekts auf dem bestimmten Objekttyp basiert.

[0062] Nach Bestimmen eines Objekt-Untertyps wird in Schritt S13 ein Teil des erzeugten Merkmalsatzes basierend auf dem bestimmten Objekt-Untertyp modifiziert. Schließlich werden die objektorientierten Bilddaten in Schritt S14 basierend auf dem modifizierten Merkmalsatz wiedergegeben.

[0063] Fig. 7 veranschaulicht weiter ein System, das die Verarbeitung von objektorientierten Bilddaten zeigt, die typisiert und untertypisiert wurden. Wein Fig. 7 gezeigt, werden die neutralen Farbbilddaten in getrennte Komponenten zerlegt. Das heißt, eine Neutral-/Nicht-Neutral-Parser-Schaltung **100** empfängt einen ankommenden Strom von Quads, die Farbbilddaten, Farbraumdaten, Objekttypdaten und Objekt-Untertypdaten enthalten. Diese Quads werden durch die Neutral-/Nicht-Neutral-Parser-Schaltung **100** in zwei Sätze von Quads zerlegt, wobei ein Satz von Quads neutrale Farben und der andere Satz von Quads nicht-neutrale Farben enthält. Der Satz, der nicht-neutrale Farben enthält, wird in der üblichen Weise durch ein Abbildungssystem **105** verarbeitet.

[0064] Andererseits werden die Quads, die die Neutralfarben-Bilddatentrios (neutrale Farbbilddaten, Farbraumdaten, Objekttypdaten und Objekt-Untertypdaten) enthaften, in eine neutrale Parser-Schaltung **101** geführt, die die ankommenden Quads basierend auf dem Merkmalsatz weiter in drei getrennte Sätze von Quads zerlegt.

[0065] Ein Quadsatz enthält Schwarzfarben-Quads, ein zweiter Quadsatz enthält Graufarben-Quads, und ein dritter Quadsatz enthält Weißfarben-Quads, wobei jeder Quad Farbbilddaten, Farbraumdaten, Objekttypdaten und Objekt-Untertypdaten enthält. Die resultierenden zerlegten Quads werden dann entsprechend dem Merkmalsatz unabhängig verarbeitet. Das heißt, die Quads, die den Schwarzfarben-Bilddaten entsprechen, werden durch einen Schwarzprozessor **102** verarbeitet, während die Quads mit den Graufarben-Bilddaten durch einen Grausprozessor **102** verarbeitet werden.

[0066] Schließlich werden die Quads, die die Weißfarben-Bilddaten enthalten, durch einen Weißprozessor **104** verarbeitet. Der Schwarzprozessor **102**, der Grauprozessor **103** und der Weißprozessor **104** sind in einer

Neutralfarben-Verarbeitungsschaltung **110** enthaften. Die verarbeiteten Neutralfarbendaten werden dann in das Abbildungssystem **105** geführt, wo sie durch den Rest des Systems in der herkömmlichen Weise verarbeitet werden, um eine Ausgabe zu erzeugen, die die richtigen Neutralwiedergabeanforderungen erfüllt.

[0067] Wie oben beschrieben, stellt die vorliegende Erfindung ein System und Verfahren bereit, um eine gewünsche neutrale Wiedergabe in einer Weise zu erlangen, die die Farbtreue nicht opfert und von den Wirkungen von Drucktreibern und Anwendungen auf neutrales Wiedergeben unabhängig ist. Die vorliegende Erfindung erlaubt ferner objektorientiertes neutrales Wiedergeben, kann benutzt werden, um die Neutralwiedergabeanforderungen von Systemen zu erfüllen, die Experten-Farbjustierungen durchführen, und stellt ein Rahmenwerk für Merkmale bereit, die neutrales Wiedergeben beeinflussen. Weiter stellt die vorliegende Erfindung ein System und Verfahren bereit, die objektorientiertes Wiedergeben unter Verwendung er es Objekttyp- und Objekt-Untertyp-Markienrungsverfahrens optimieren können.

[0068] Obwohl die vorliegende Erfindung oben im Einzelnen beschrieben wurde, können verschiedene Modifikationen implementiert werden, ohne von der vorliegenden Erfindung abzuweichen. Zum Beispiel ist die bevorzugte Ausführung der vorliegenden Erfindung in Bezug auf ein Drucksystem beschrieben worden, aber dieses Verfahren ist ohne weiteres in einem Anzeigesystem zu implementieren. Des Weiteren ist die vorliegende Erfindung nicht auf ein xerographisches System begrenzt, sondern ist unabhängig von dem Ausgabesystem und kann leicht in einer Tintenstrahl-Umgebung verwendet werden. Es ist ebenfalls auf Hifi-Farbsysteme anwendbar.

Patentansprüche

1. System zur Verarbeitung objektorientierter Bilddaten, das umfasst:
eine erste Parser-Schaltung (**100**), um die objektorientierten Bilddaten in nicht-neutrale Bilddaten und neutrale Bilddaten zu zerlegen; und **dadurch gekennzeichnet**, dass das System des Weiteren umfasst:
eine zweite Parser-Schaltung (**101**), um die neutralen Bilddaten in Scharzbilddaten, Graubilddaten und Weißbilddaten zu zerlegen, und
eine Neutralfarben-Verarbeitungsschaltung (**110**), um die Scharzbilddaten, Graubilddaten und die Weißbilddaten zu verarbeiten.

2. System nach Anspruch 1, wobei die Neutralfarben-Verarbeitungsschaltung (**110**) umfasst:
eine Schwarzverarbeitungsschaltung (**102**), um die Schwarzbilddaten zu verarbeiten;
eine Grauverarbeitungsschaltung (**103**), um die Graubilddaten zu verarbeiten, und
eine Weißverarbeitungsschaltung (**104**), um die Weißbilddaten zu verarbeiten.

3. System nach Anspruch 1 oder Anspruch 2, wobei die Neutralfarben-Verarbeitungsschaltung (**110**) die Schwarz-, Grau- und Weißbilddaten entsprechend einem ausgewählten Merkmalsatz verarbeitet.

4. Verfahren zur Verarbeitung objektorientierter Bilddaten, wobei das Verfahren die folgenden Schritte umfasst:
(a) Zerlegen der objektorientierten Bilddaten in nicht-neutrale Bilddaten und neutrale Bilddaten;
(b) Zerlegen der neutralen Bilddaten in Scharzbilddaten, Graubilddaten und Weißbilddaten;
(c) Verarbeiten der Scharzbilddaten, der Graubilddaten und der Weißbilddaten, und
(d) Verarbeiten der verarbeiteten Schwarzbilddaten, der verarbeiteten Graubilddaten, der verarbeiteten Weißbilddaten und der nicht-neutralen Bilddaten.

5. Verfahren nach Anspruch 4, wobei der Schritt (c) die Schwarz-, Grau- und Weißbilddaten entsprechend einem ausgewählten Merkmalsatz verarbeitet.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

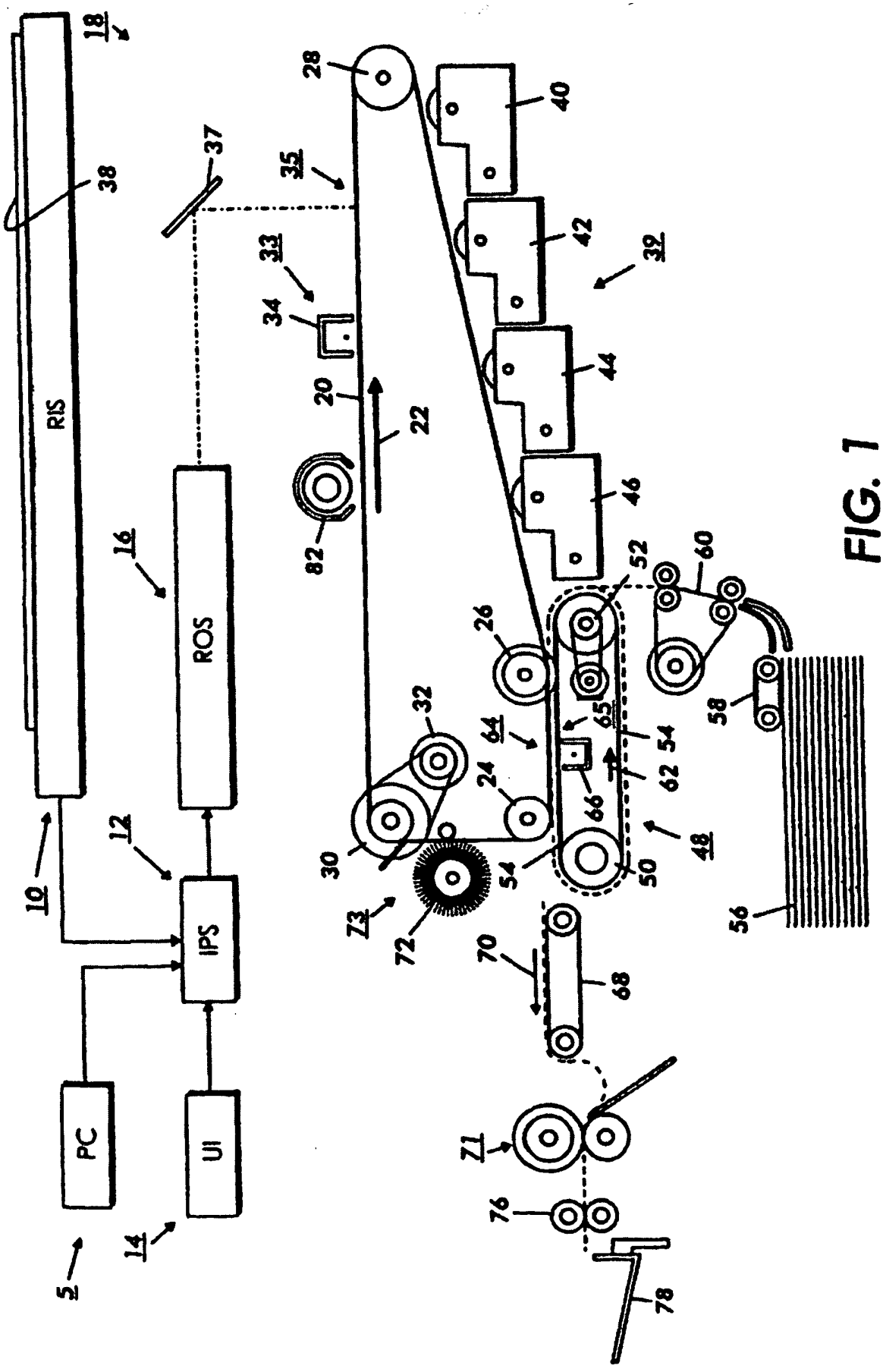

FIG. 1

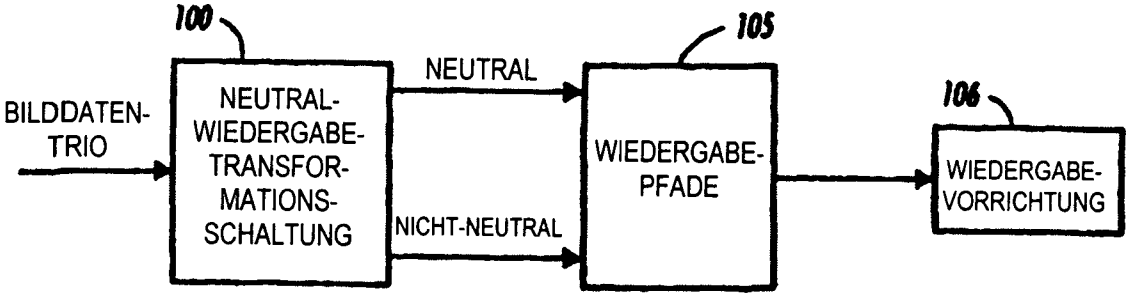
100
BILDDATEN-TRIO
NEUTRAL-WIEDERGABE-TRANSFOR-MATIONS-SCHALTUNG
NEUTRAL
NICHT-NEUTRAL
105
WIEDERGABE-PFADE
106
WIEDERGABE-VORRICHTUNG

FIG. 2

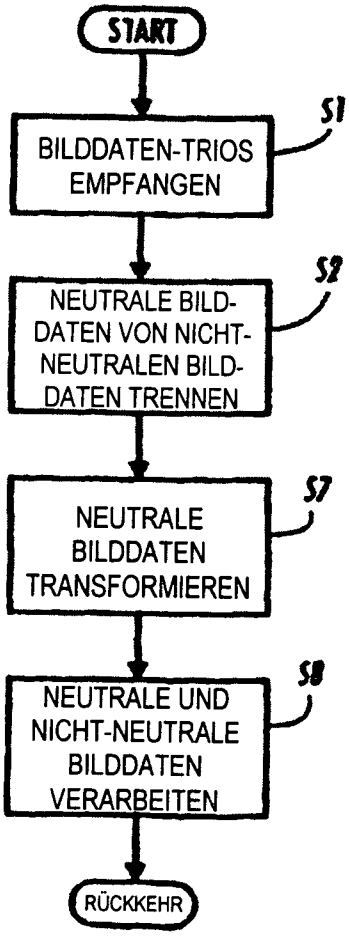
START
S1
BILDDATEN-TRIOS EMPFANGEN
S2
NEUTRALE BILD-DATEN VON NICHT-NEUTRALEN BILD-DATEN TRENNEN
S7
NEUTRALE BILDDATEN TRANSFORMIEREN
S8
NEUTRALE UND NICHT-NEUTRALE BILDDATEN VERARBEITEN
RÜCKKEHR

FIG. 3

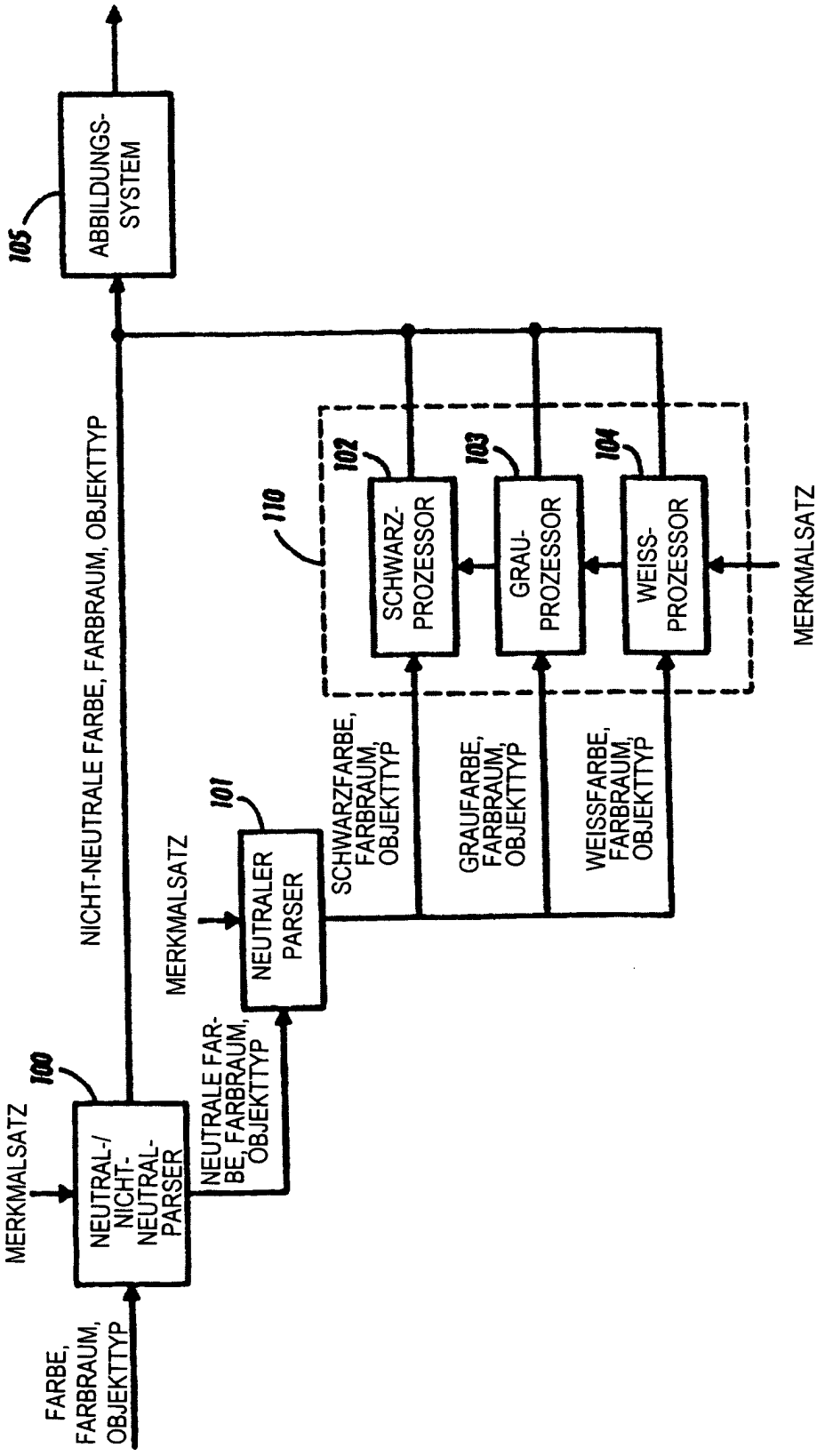
FARBE, FARBRAUM, OBJEKTTYP
MERKMALSATZ
100
NEUTRAL-/ NICHT-NEUTRAL-PARSER
NEUTRALE FAR-BE, FARBRAUM, OBJEKTTYP
MERKMALSATZ
101
NEUTRALER PARSER
SCHWARZFARBE, FARBRAUM, OBJEKTTYP
GRAUFARBE, FARBRAUM, OBJEKTTYP
WEISSFARBE, FARBRAUM, OBJEKTTYP
110
102
SCHWARZ-PROZESSOR
103
GRAU-PROZESSOR
104
WEISS-PROZESSOR
MERKMALSATZ
NICHT-NEUTRALE FARBE, FARBRAUM, OBJEKTTYP
105
ABBILDUNGS-SYSTEM

FIG. 4

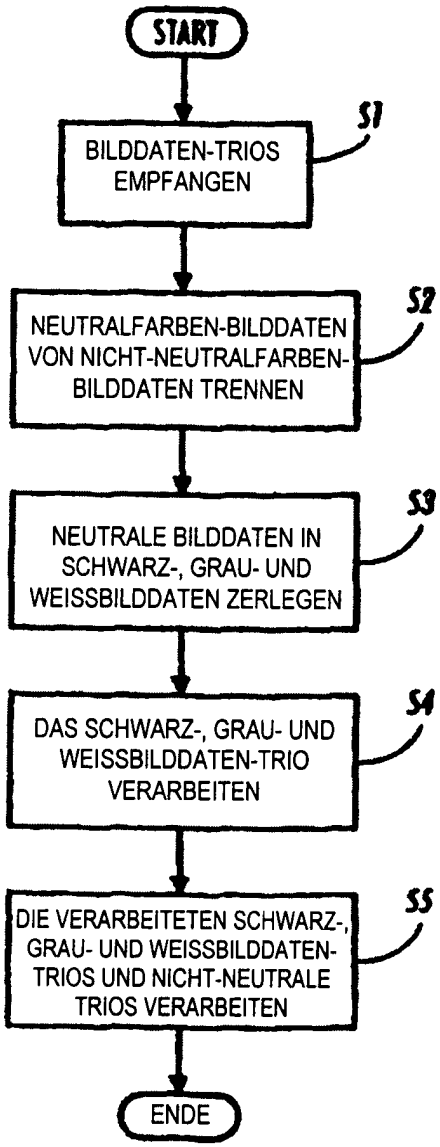
START
BILDDATEN-TRIOS EMPFANGEN
S1
NEUTRALFARBEN-BILDDATEN VON NICHT-NEUTRALFARBEN-BILDDATEN TRENNEN
S2
NEUTRALE BILDDATEN IN SCHWARZ-, GRAU- UND WEISSBILDDATEN ZERLEGEN
S3
DAS SCHWARZ-, GRAU- UND WEISSBILDDATEN-TRIO VERARBEITEN
S4
DIE VERARBEITETEN SCHWARZ-, GRAU- UND WEISSBILDDATEN-TRIOS UND NICHT-NEUTRALE TRIOS VERARBEITEN
S5
ENDE

FIG. 5

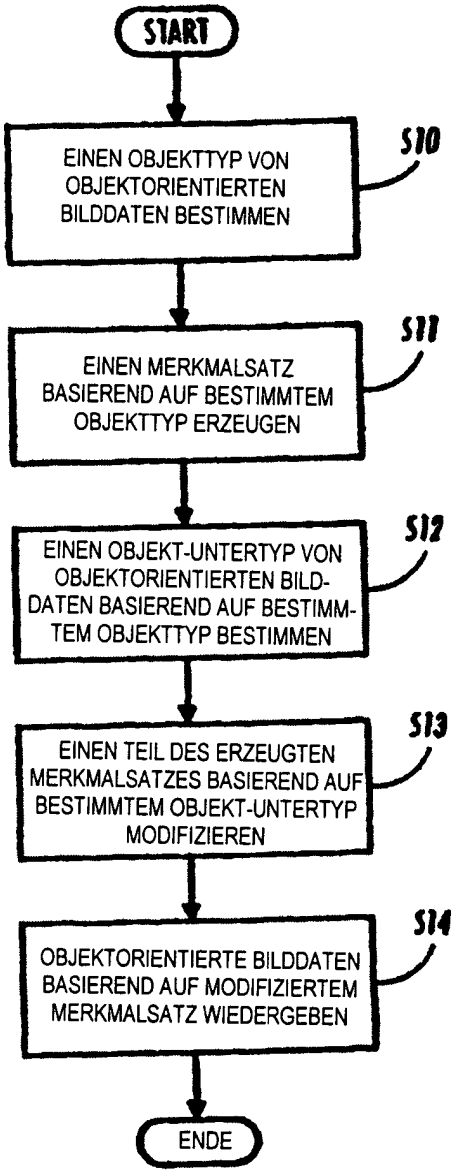
START
EINEN OBJEKTTYP VON OBJEKTORIENTIERTEN BILDDATEN BESTIMMEN
S10
EINEN MERKMALSATZ BASIEREND AUF BESTIMMTEM OBJEKTTYP ERZEUGEN
S11
EINEN OBJEKT-UNTERTYP VON OBJEKTORIENTIERTEN BILD-DATEN BASIEREND AUF BESTIMM-TEM OBJEKTTYP BESTIMMEN
S12
EINEN TEIL DES ERZEUGTEN MERKMALSATZES BASIEREND AUF BESTIMMTEM OBJEKT-UNTERTYP MODIFIZIEREN
S13
OBJEKTORIENTIERTE BILDDATEN BASIEREND AUF MODIFIZIERTEM MERKMALSATZ WIEDERGEBEN
S14
ENDE

FIG. 6

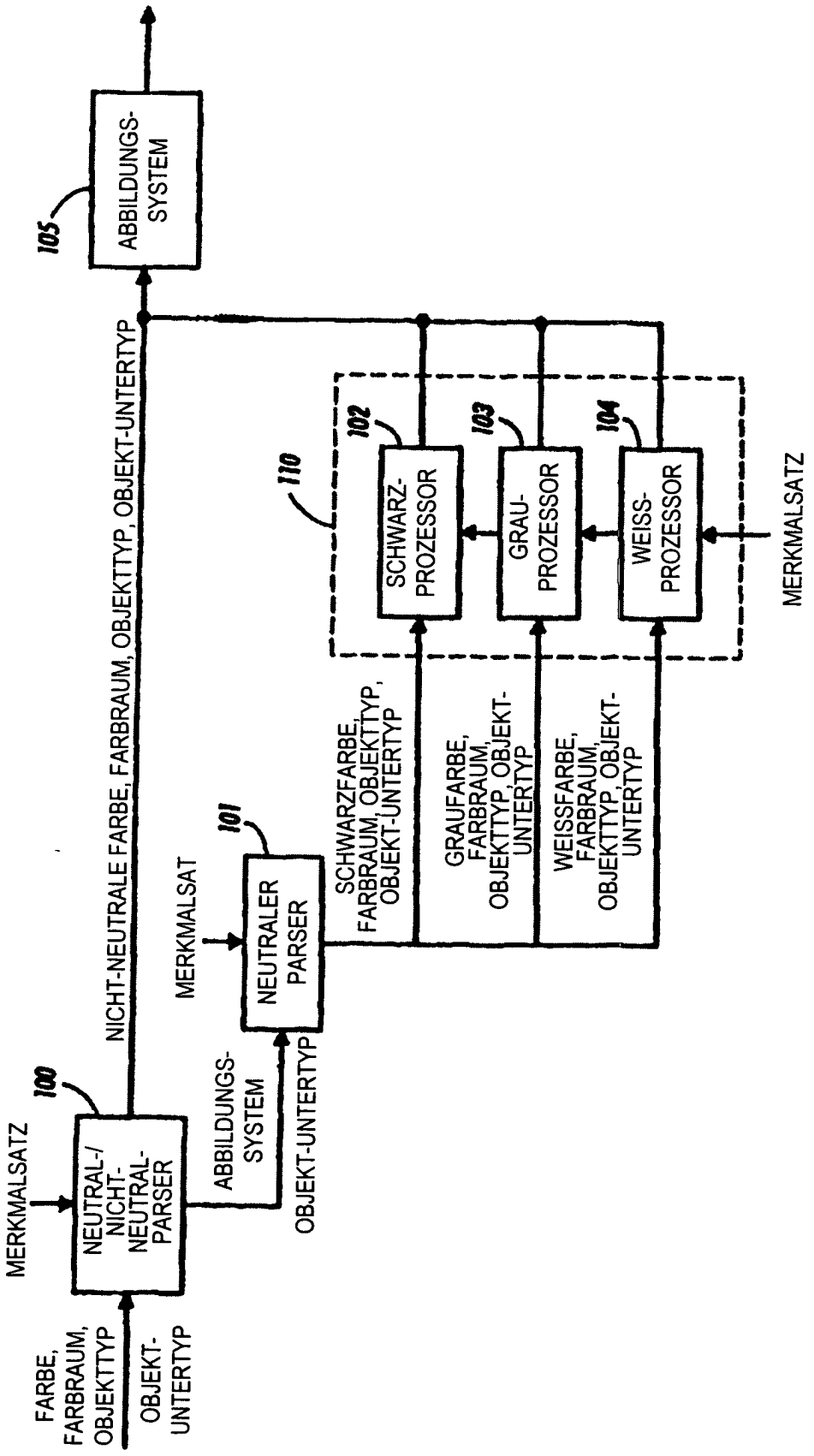
FARBE, FARBRAUM, OBJEKTTYP
OBJEKT-UNTERTYP
MERKMALSATZ
100
NEUTRAL-/ NICHT-NEUTRAL-PARSER
NICHT-NEUTRALE FARBE, FARBRAUM, OBJEKTTYP, OBJEKT-UNTERTYP
ABBILDUNGS-SYSTEM
OBJEKT-UNTERTYP
MERKMALSAT
101
NEUTRALER PARSER
SCHWARZFARBE, FARBRAUM, OBJEKTTYP, OBJEKT-UNTERTYP
GRAUFARBE, FARBRAUM, OBJEKTTYP, OBJEKT-UNTERTYP
WEISSFARBE, FARBRAUM, OBJEKTTYP, OBJEKT-UNTERTYP
110
102
SCHWARZ-PROZESSOR
103
GRAU-PROZESSOR
104
WEISS-PROZESSOR
MERKMALSATZ
105
ABBILDUNGS-SYSTEM

FIG. 7

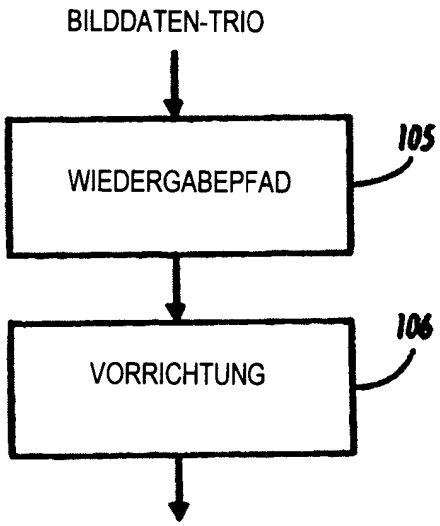

FIG. 8
(Stand der Technik)