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

(10) **DE 60 2004 002 180 T2** 2007.07.12

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 542 155 B1**
(21) Deutsches Aktenzeichen: **60 2004 002 180.6**
(96) Europäisches Aktenzeichen: **04 256 874.1**
(96) Europäischer Anmeldetag: **05.11.2004**
(97) Erstveröffentlichung durch das EPA: **15.06.2005**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **30.08.2006**
(47) Veröffentlichungstag im Patentblatt: **12.07.2007**

(51) Int Cl.[8]: ***G06K 9/62*** (2006.01)
G06K 9/00 (2006.01)
G06T 7/00 (2006.01)

(30) Unionspriorität:
0328736 **11.12.2003** **GB**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Sony United Kingdom Ltd., Brooklands, Weybridge, GB

(74) Vertreter:
Mitscherlich & Partner, Patent- und Rechtsanwälte, 80331 München

(72) Erfinder:
Haynes, Simon Dominic, Basingtoke, Hampshire RG21 3JX, GB; Living, Jonathan, West Midlands DY8 5HZ, GB

(54) Bezeichnung: **Objekterkennung**

Beschreibung

[0001] Diese Erfindung betrifft eine Objekterkennung.

[0002] Die folgende Beschreibung betrifft ein Problem, das bei der Erfassung verschiedener Arten von Objekten vorhanden ist, aber zwecks Klarheit der Beschreibung bezüglich einer Gesichtserkennung diskutiert wird.

[0003] Es wurden viele Algorithmen zur Erkennung menschlicher Gesichter in der Literatur vorgeschlagen, einschließlich der Verwendung von so genannten Eigengesichtern, Gesichtsschablonengleichheitsprüfung, verformbarer Schablonengleichheitsprüfung oder neuronaler Netzklassifizierung. Keine von diesen ist perfekt, und jede hat im Allgemeinen zugehörige Vorteile und Nachteile. Keine gibt eine absolut zuverlässige Angabe, dass ein Bild ein Gesicht enthält; im Gegenteil basieren sie alle auf einer probalistischen Beurteilung basierend auf einer mathematischen Analyse des Bildes, ob das Bild wenigstens eine gewisse Wahrscheinlichkeit des Enthaltens eines Gesichts besitzt. Zum Beispiel offenbart C.C. Chang et al., "A color image retrieval method based on local histogram", Proc. IEEE Pacific Rim Conf. on mulitimedia information processing, Seiten 831–836, 2001, einen Bildvergleich basierend auf blockweisen Farbhistogrammen, während B. Heisele et al., "Facial recognition: Component-based versus global approaches", Computer vision and image understanding, Vol. 91, Nr. 1–2, Seiten 6–21, eine komponenten-basierte Gegenstandserkennung offenbart. In Abhängigkeit von ihrer Anwendung haben die Algorithmen im Allgemeinen den Wahrscheinlichkeitsschwellenwert ziemlich hoch eingestellt, um falsche Erfassungen von Gesichtern zu vermeiden.

[0004] Die Objekterkennung neigt dazu, sehr prozessorintensiv zu sein. In Situationen, in denen die Objekterkennung in Echtzeit ausgeführt werden muss, kann es schwierig sein, die gesamte Objekterkennungsverarbeitung in der zulässigen Zeit – z.B. eine Vollbildperiode eines Videosignals – abzuschließen.

[0005] Diese Erfindung sieht eine Objekterkennungsvorrichtung zum Erkennen von Objekten in einem Prüfbild vor, wobei die Vorrichtung aufweist:
eine Einrichtung zum Vergleichen von Blöcken eines Prüffensters des Bildes mit Referenzdaten, die die Anwesenheit eines Objekts anzeigen, um Indexpunktzahlen zu erzeugen, die einen Ähnlichkeitsgrad zwischen einem Bereich und den Referenzdaten angeben;
eine Einrichtung zum Speichern von Wahrscheinlichkeitsdaten entsprechend möglichen Werten der Indexpunktzahl und der Blockposition;
eine Einrichtung, die bezüglich eines aktuellen Blocks zum Zugreifen auf einen Wahrscheinlichkeitswert aus der Speichereinrichtung in Abhängigkeit von dieser Blockposition im Prüffenster und der bezüglich dieses Blocks erzeugten Indexpunktzahl ausgebildet ist; und
eine Einrichtung zum Kombinieren der Wahrscheinlichkeitswerte entsprechend Blöcken in einem Prüffenster, um ein Ergebnis zu erzeugen, das die Wahrscheinlichkeit anzeigt, dass dieses Prüffenster ein Objekt enthält,
gekennzeichnet durch die Zugriffseinrichtung, die ausgebildet ist, um auf zwei oder mehr Wahrscheinlichkeitswerte bezüglich einer aktuellen Blockposition und Indexpunktzahl zuzugreifen, wobei die zwei oder mehr Wahrscheinlichkeitswerte sich auf verschiedene Objektorientierungen beziehen.

[0006] Die Erfindung beinhaltet die Erkenntnis, auf welche Werte nebeneinander in einem Objekterkennungsprozess zugegriffen wird, wobei diese Werte nebeneinander oder nahe beieinander geordnet sind, um die Effizienz des Datenzugriffsvorgangs zu verbessern.

[0007] Verschiedene weitere jeweilige Aspekt und Merkmale der Erfindung sind in den anhängenden Ansprüchen definiert.

[0008] Es werden nun Ausführungsbeispiele der Erfindung nur beispielhaft unter Bezug auf die beiliegenden Zeichnungen beschrieben, in denen gleiche Teile durch gleiche Ziffern definiert sind. Darin zeigen:

[0009] Fig. 1 eine schematische Darstellung eines Mehrzweck-Computersystems zur Verwendung als ein Gesichtserkennungssystem und/oder ein nicht-lineares Editiersystem;

[0010] Fig. 2 eine schematische Darstellung einer einen Videokamerarekorder (Camcorder) benutzenden Gesichtserkennung;

[0011] **Fig.** 3 eine schematische Darstellung, die einen Trainingsprozess veranschaulicht;

[0012] Fig. 4 eine schematische Darstellung, die einen Erkennungsprozess veranschaulicht;

[0013] Fig. 5 schematisch ein Merkmalshistogramm;

[0014] Fig. 6 schematisch einen Abtastprozess, um Eigenblöcke zu erzeugen;

[0015] Fig. 7 und Fig. 8 schematisch Sätze von Eigenblöcken;

[0016] Fig. 9 schematisch einen Prozess zum Bilden eines eine Blockposition darstellenden Histogramms;

[0017] Fig. 10 schematisch die Erzeugung einer Histogrammlagernummer;

[0018] Fig. 11 schematisch der Berechnung einer Gesichtswahrscheinlichkeit;

[0019] Fig. 12a bis Fig. 12f schematische Beispiele von Histogrammen, die mittels der obigen Verfahren erzeugt wurden;

[0020] Fig. 13a und Fig. 13b schematisch die Datenstruktur der Histogramme;

[0021] Fig. 14 schematisch eine so genannte Lagerkarte mit einem darüber gelegten Gesichtsfenster;

[0022] Fig. 15a bis Fig. 15g schematisch eine so genannte Mehrmaßstabs-Gesichtserkennung;

[0023] Fig. 16 ein schematisches Flussdiagramm, das eine Technik zum Erfassen von Gesichtspositionen in einer Mehrmaßstabsanordnung veranschaulicht;

[0024] Fig. 17 schematisch einen Bewegungsdetektor;

[0025] Fig. 18a bis Fig. 18e schematisch eine Technik zum Erfassen eines Bereichs einer Veränderung in einem Bild;

[0026] Fig. 19a bis Fig. 19c schematisch eine Verbesserung der Technik von Fig. 18a bis Fig. 18e;

[0027] Fig. 20a bis Fig. 20c schematisch eine Raumdezimierungstechnik;

[0028] Fig. 21a bis Fig. 21d schematisch eine weitere Raumdezimierungstechnik;

[0029] Fig. 22 schematisch einen Gesichtsverfolgungsalgorithmus;

[0030] Fig. 23a und Fig. 23b schematisch die Ableitung eines für eine Hautfarbenerkennung benutzten Suchbereichs;

[0031] Fig. 24 schematisch eine bei der Hautfarbenerkennung angewendete Maske;

[0032] Fig. 25a bis Fig. 25c schematisch die Verwendung der Maske von Fig. 24;

[0033] Fig. 26 eine schematische Entfernungsabbildung;

[0034] Fig. 27 schematisch einen Farbmaskenprozess;

[0035] Fig. 28 schematisch einen Farbabbildungsaktualisierungsprozess; und

[0036] Fig. 29a bis Fig. 29c schematisch eine Gradienten-(Varianz-)Vorverarbeitungstechnik.

[0037] Die Ausführungsbeispiele werden bezüglich einer Gesichtserkennung beschrieben, aber sie sind durch Trainieren mit die erforderlichen Objekte darstellenden Trainingsbildern gleichermaßen auf eine Erkennung anderer Objekte, wie beispielsweise Autos, anwendbar.

[0038] Fig. 1 ist eine schematische Darstellung eines Mehrzweck-Computersystems zur Verwendung als ein Gesichtserkennungssystem und/oder ein nicht-lineares Editiersystem. Das Computersystem weist eine Verar-

beitungseinheit **10** (neben weiteren herkömmlichen Komponenten) mit einer Zentraleinheit (CPU) **20**, einem Speicher wie beispielsweise einem Direktzugriffsspeicher (RAM) **30** und einem nichtflüchtigen Speicher wie beispielsweise einem Plattenlaufwerk **40** auf. Das Computersystem kann mit einem Netzwerk **50**, wie beispielsweise einem lokalen Netz oder dem Internet (oder mit beiden) verbunden werden. Eine Tastatur **60**, eine Maus oder eine andere Nutzereingabevorrichtung **70** und ein Anzeigeschirm **80** sind ebenfalls vorgesehen. Der Fachmann wird erkennen, dass ein Mehrzweck-Computersystem viele weitere Teile enthalten kann, die hier nicht beschrieben werden müssen.

[0039] Fig. 2 ist eine schematische Darstellung einer einen Videokamerarecorder (Camcorder) benutzenden Gesichtserkennung. Der Camcorder **100** weist eine Linse **110** auf, die ein Bild auf eine Bilderfassungsvorrichtung **120** mit ladungsgekoppelter Schaltung (CCD) fokussiert. Das resultierende Bild in elektronischer Form wird durch eine Bildverarbeitungslogik **130** zum Aufzeichnen auf ein Aufzeichnungsmedium wie beispielsweise eine Bandkassette **140** verarbeitet. Die durch die Vorrichtung **120** erfassten Bilder werden auch auf einer Nutzeranzeige **150** angezeigt, die durch ein Okular **160** betrachtet werden kann.

[0040] Um mit den Bildern zusammenhängende Töne zu erfassen, werden ein oder mehrere Mikrofone benutzt. Diese können in dem Sinn, dass sie mit dem Camcorder durch ein flexibles Kabel verbunden sind, externe Mikrofone sein oder sie können an dem Camcordergehäuse selbst montiert sein. Analoge Audiosignale von dem/den Mikrofon(en) werden durch eine Audioverarbeitungsanordnung **170** verarbeitet, um entsprechende Audiosignale zur Aufzeichnung auf dem Speichermedium **140** zu erzeugen.

[0041] Es ist anzumerken, dass die Video- und Audiosignale auf dem Speichermedium **140** in entweder digitaler Form oder analoger Form oder sogar in beiden Formen aufgezeichnet werden können. Somit können die Bildverarbeitungsanordnung **130** und die Audioverarbeitungsanordnung **170** eine Stufe einer Analog/Digital-Umsetzung enthalten.

[0042] Der Camcorderbenutzer ist in der Lage, Aspekte des Leistungsvermögens der Linse **110** durch Benutzersteuerungen **180** zu steuern, die eine Linsensteueranordnung **190** beeinflussen, um elektrische Steuersignale **200** zur Linse **110** zu schicken.

[0043] Typischerweise werden auf diese Weise Attribute, wie beispielsweise Fokus und Zoom, gesteuert, aber die Blende und weitere Attribute können ebenfalls durch den Benutzer gesteuert werden.

[0044] Zwei weitere Benutzersteuerungen sind schematisch dargestellt. Ein Druckknopf **210** ist vorgesehen, um eine Aufzeichnung auf das Aufzeichnungsmedium **140** zu starten und zu stoppen. Zum Beispiel kann ein Drücken der Steuerung **210** die Aufzeichnung starten und ein weiteres Drücken kann die Aufzeichnung stoppen, oder die Steuerung muss für die Aufzeichnung in einem gedrückten Zustand gehalten werden, oder ein Drücken kann die Aufzeichnung für eine bestimmte Zeitdauer von zum Beispiel fünf Sekunden starten. Bei jeder dieser Anordnungen ist es technisch sehr einfach, den Aufzeichnungsbetrieb des Camcorders einzurichten, wo der Beginn und das Ende jeder „Aufnahme" (fortlaufende Dauer einer Aufzeichnung) stattfindet.

[0045] Die andere schematisch in Fig. 2 dargestellte Benutzersteuerung ist ein „Markierer für gute Aufnahmen" (GSM) **220**, der durch den Benutzer betätigt werden kann. Um „Metadaten" (zugehörige Daten) in Zusammenhang mit dem Video- und Audiomaterial auf dem Aufzeichnungsmedium **140** speichern zu lassen, die anzeigen, dass diese spezielle Aufnahme durch den Benutzer in gewisser Beziehung (zum Beispiel waren die Akteure besonders gut; der Nachrichtenreporter betonte jedes Wort korrekt; usw.) subjektiv als „gut" angesehen wurde.

[0046] Die Metadaten können in einer bestimmten ungenutzten Kapazität (z. B. „Nutzerdaten") auf dem Aufzeichnungsmedium **140** aufgezeichnet werden, in Abhängigkeit von dem speziellen gebrauchten Format und Standard. Alternativ können die Metadaten auf einem separaten Speichermedium wie beispielsweise einem entfernbaren MemoryStick™-Speicher (nicht dargestellt) gespeichert werden, oder die Metadaten können in einer externen Datenbank (nicht dargestellt) gespeichert werden, indem sie zum Beispiel über eine drahtlose Verbindung (nicht dargestellt) zu einer solchen Datenbank kommuniziert werden. Die Metadaten können nicht nur die GSM-Informationen enthalten, sondern auch Aufnahmegrenzen, Linsenattribute, durch einen Benutzer (z.B. an einer Tastatur – nicht dargestellt) eingegebene alphanumerische Informationen, geographische Positionsinformationen von einem GPS-Empfänger (nicht dargestellt) und dergleichen.

[0047] Bisher deckte die Beschreibung einen metadaten-fähigen Camcorder ab. Es wird nun der Weg beschrieben, wie eine Gesichtserkennung auf einen solchen Camcorder angewendet werden kann. Es ist natür-

lich selbstverständlich, dass die Techniken auf zum Beispiel eine Netzwerkkamera wie beispielsweise eine Internetprotokoll (IP) – Kamera, eine Videokonferenzkamera und dergleichen anwendbar sind.

[0048] Der Camcorder enthält eine Gesichtsdetektoranordnung **230**. Entsprechende Anordnungen werden unten in mehr Einzelheiten beschrieben, aber für diesen Teil der Beschreibung ist es ausreichend, zu sagen, dass die Gesichtsdetektoranordnung **230** Bilder von der Bildverarbeitungsanordnung **130** empfängt und erfasst oder zu erfassen versucht, ob solche Bilder ein oder mehrere Gesichter enthalten. Der Gesichtsdetektor kann Gesichtserkennungsdaten ausgeben, die in der Form eines „Ja/Nein"-Zeichens vorliegen können oder darin detaillierter sein können, dass die Daten die Bildkoordinaten der Gesichter, wie beispielsweise die Koordinaten von Augenpositionen in jedem erkannten Gesicht, enthalten können. Diese Informationen können als eine weitere Art von Metadaten behandelt und in irgendeinem der anderen oben beschriebenen Formate gespeichert werden.

[0049] Wie nachfolgend beschrieben, kann die Gesichtserkennung durch Verwendung weiterer Arten von Metadaten in dem Erkennungsprozess unterstützt werden. Zum Beispiel empfängt der Gesichtsdetektor **230** ein Steuersignal von der Linsensteueranordnung **190**, um die aktuellen Fokus- und Zoomeinstellungen der Linse **110** anzuzeigen. Diese können den Gesichtsdetektor unterstützen, indem sie eine Ausgangsanzeige der erwarteten Bildgröße von Gesichtern, die im Vordergrund des Bildes vorhanden sein könne, geben. Diesbezüglich ist anzumerken, dass die Fokus- und die Zoomeinstellungen zwischen ihnen die erwartete Trennung zwischen dem Camcorder **100** und einer gefilmten Person und auch die Vergrößerung der Linse **110** definieren. Aus diesen zwei Attributen ist es basierend auf einer durchschnittlichen Gesichtsgröße möglich, die erwartete Größe (in Pixeln) eines Gesichts in den resultierenden Bilddaten zu berechnen.

[0050] Ein herkömmlicher (bekannter) Sprachdetektor **240** empfängt Audioinformationen von der Audioverarbeitungsanordnung **170** und erfasst das Vorhandensein von Sprache in solchen Audioinformationen. Das Vorhandensein von Sprache kann ein Indikator sein, dass die Wahrscheinlichkeit eines in den entsprechenden Bildern vorhanden Gesichts höher ist als wenn keine Sprache erkannt wird. In manchen nachfolgend zu erläuternden Ausführungsbeispielen kann der Sprachdetektor so modifiziert werden, dass er eine Lokalisierung eines Sprechers durch Erfassen eines aktivsten Mikrofons aus einem Satz von Mikrofonen oder durch eine Triangulation oder eine ähnliche Technik zwischen mehreren Mikrofonen vorsieht.

[0051] Schließlich werden die GSM-Informationen **220** und die Aufnahmeinformationen (von der Steuerung **210**) dem Gesichtsdetektor **230** zugeführt, um Aufnahmegrenzen und die durch den Benutzer am nützlichsten angesehenen Aufnahmen anzuzeigen.

[0052] Natürlich können, falls der Camcorder auf der analogen Aufzeichnungstechnik basiert, weitere Analog/Digital-Umsetzer (ADCs) erforderlich sein, um die Bild- und Audioinformationen abzuwickeln.

[0053] Das vorliegende Ausführungsbeispiel verwendet eine Gesichtserkennungstechnik, die als zwei Phasen angeordnet ist. **Fig.** 3 ist eine schematische Darstellung einer Trainingsphase, und **Fig. 4** ist eine schematische Darstellung einer Erkennungsphase.

[0054] Anders als manche früher vorgeschlagenen Gesichtserkennungsverfahren (siehe Literaturhinweise 4 und 5 unten) basiert das vorliegende Verfahren auf dem Modellieren des Gesichts in Teilen anstelle als Ganzes. Die Teile können entweder über die angenommenen Positionen der Gesichtsmerkmale zentrierte Blöcke (so genanntes „selektives Abtasten") oder in regelmäßigen Abständen über das Gesicht abgetastete Blöcke (so genanntes „regelmäßiges Abtasten") sein. Die vorliegende Beschreibung deckt hauptsächlich das regelmäßige Abtasten ab, da in empirischen Tests herausgefunden wurde, dass dies die besseren Ergebnisse ergibt.

[0055] In der Trainingsphase wird ein Analysevorgang auf einen Satz von Bildern, von denen man weiß, dass sie Gesichter enthalten, und (optional) einen weiteren Satz von Bildern („Nichtgesichtsbilder"), von denen man weiß, dass sie keine Gesichter enthalten, angewendet. Der Analyseprozess bildet ein mathematisches Modell von Gesichts- und Nichtgesichtsmerkmalen, mit dem später ein Prüfbild verglichen werden kann (in der Erkennungsphase).

[0056] Somit sind zum Aufbauen des mathematischen Modells (Trainingsprozess **310** von **Fig.** 3) die Grundschritte wie folgt:

1. Aus einem Satz **300** von Gesichtsbildern, die normiert sind, um die gleichen Augenpositionen zu haben, wird jedes Gesicht regelmäßig in kleine Blöcke abgetastet.

2. Attribute werden für jeden Block berechnet; diese Attribute werden weiter unten erläutert.
3. Die Attribute werden auf eine verwaltbare Anzahl von unterschiedlichen Werten Quantisiert.
4. Die quantisierten Attribute werden dann kombiniert, um einen einzelnen quantisierten Wert bezüglich dieser Blockposition zu erzeugen.
5. Der einzelne quantisierte Wert wird dann als ein Eintrag in einem Histogramm, wie beispielsweise dem schematischen Histogramm von Fig. 5, aufgezeichnet. Die kollektiven Histogramminformationen **320** bezüglich aller Blockpositionen in allen Trainingsbildern bilden das Fundament des mathematischen Modells der GesichtsMerkmale.

[0057] Ein solches Histogramm wird für jede mögliche Blockposition durch Wiederholen der obigen Schritte bezüglich einer großen Anzahl von Prüfgesichtsbildern vorbereitet. Die Testdaten werden in einem Anhang A unten weiter beschrieben. Somit werden in einem System, das eine Anordnung von 8 × 8 Blöcken benutzt, 64 Histogramme erstellt. In einem späteren Teil der Verarbeitung wird ein quantisiertes Prüfattribut mit den Histogrammdaten verglichen; die Tatsache, dass ein Gesamthistogramm benutzt wird, um die Daten zu modellieren, bedeutet, dass keine Annahmen gemacht werden müssen, ob es einer parametrisierten Verteilung, z. B. Gauß oder anders, folgt. Um (erforderlichenfalls) Datenspeicherplatz zu sparen, können Histogramme, die ähnlich sind, verbunden werden, sodass das gleiche Histogramm für unterschiedliche Blockpositionen wiederverwendet werden kann.

[0058] In der Erkennungsphase werden, um den Gesichtsdetektor auf ein Prüfbild **350** anzuwenden, aufeinanderfolgende Fenster in dem Prüfbild wie folgt verarbeitet **340**:

6. Das Fenster wird als eine Reihe von Blöcken regelmäßig abgetastet, und Attribute bezüglich jedes Blocks werden wie in den Stufen 1 bis 4 oben berechnet und quantisiert.
7. Entsprechende „Wahrscheinlichkeiten" für die quantisierten Attributwerte für jede Blockposition werden aus den entsprechenden Histogrammen abgerufen. Das heißt, für jede Blockposition wird ein jeweiliges quantisiertes Attribut erzeugt und mit einem bezüglich dieser Blockposition im Voraus erzeugten Histogramm verglichen. Die Art, in welcher die Histogramme zu „Wahrscheinlichkeitsdaten" führen, wird nachfolgend beschrieben.
8. Alle oben erhaltenen Wahrscheinlichkeiten werden miteinander multipliziert, um eine Endwahrscheinlichkeit zu bilden, die mit einem Schwellenwert verglichen wird, um das Fenster als „Gesicht" oder „kein Gesicht" zu klassifizieren. Es ist selbstverständlich, dass das Erkennungsergebnis „Gesicht" oder „kein Gesicht" eher eine wahrscheinlichkeitsbasierte Messung als eine absolute Erkennung ist. Manchmal kann ein „kein Gesicht" enthaltendes Bild fälschlicherweise als „Gesicht" erkannt werden, ein so genanntes falsches Positivergebnis. In anderen Fällen kann ein ein Gesicht enthaltendes Bild fälschlicherweise als „kein Gesicht" erkannt werden, ein so genanntes falsches Negativergebnis. Es ist ein Ziel jedes Gesichtserkennungssystems, den Anteil falscher Positivergebnisse und den Anteil falscher Negativergebnisse zu reduzieren, aber es ist natürlich selbstverständlich, dass ein Reduzieren dieser Anteile auf Null mit derzeitiger Technik schwierig ist, falls nicht unmöglich.

[0059] Wie oben erwähnt kann in der Trainingsphase ein Satz von „Nichtgesichtsbildern" benutzt werden, um einen entsprechenden Satz von „Nichtgesichtshistogrammen" zu erzeugen. Dann kann, um die Erkennung eines Gesichts zu erzielen, die aus den Nichtgesichtshistogrammen erzeugte „Wahrscheinlichkeit" mit einem separaten Schwellenwert verglichen werden, sodass die Wahrscheinlichkeit unter dem Schwellenwert für das Prüffenster liegen muss, um ein Gesicht zu enthalten. Alternativ kann das Verhältnis der Gesichtswahrscheinlichkeit zur Nichtgesichtswahrscheinlichkeit mit einem Schwellenwert verglichen werden.

[0060] Extra Trainingsdaten können durch Anwenden von „synthetischen Veränderungen" **330** an dem originalen Trainingssatz erzeugt werden, wie beispielsweise Veränderung in Position, Ausrichtung, Größe, Seitenverhältnis, Hintergrundszenerie, Lichtintensität und Frequenzgehalt.

[0061] Es wird nun die Ableitung der Attribute und ihrer Quantisierung beschrieben. In der vorliegenden Technik werden Attribute bezüglich so genannter Eigenblöcke gemessen, die Kernblöcke (oder Eigenvektoren) sind, die unterschiedliche Arten von Blöcken darstellen, die in dem mit Fenstern eingeteilten Bild vorhanden sind. Die Erzeugung von Eigenblöcken wird unter Bezug auf Fig. 6 zuerst beschrieben.

Eigenblockerzeugung

[0062] Die Attribute im vorliegenden Ausführungsbeispiel basieren auf so genannten Eigenblöcken. Die Eigenblöcke wurden so konstruiert, dass sie eine gute Darstellungsfähigkeit der Blöcke im Trainingssatz haben. Deshalb wurden sie durch Durchführen einer Hauptkomponentenanalyse an einem großen Satz von Blöcken

aus dem Trainingssatz erzeugt. Dieser Prozess ist schematisch in Fig. 6 dargestellt und in mehr Einzelheiten im Anhang B beschrieben.

Training des Systems

[0063] Es wurden Versuche mit zwei unterschiedlichen Sätzen von Trainingsblöcken durchgeführt.

Eigenblocksatz I

[0064] Zunächst wurde ein Satz von Blöcken benutzt, der aus 25 Gesichtsbildern im Trainingssatz genommen wurde. Die 16×16-Blöcke wurden alle 16 Pixel abgetastet und waren daher nicht überlappend. Diese Abtastung ist in Fig. 6 dargestellt. Wie man sehen kann, werden aus jedem 64×64-Trainingsbild 16 Blöcke erzeugt. Dies führt insgesamt zu einer Anzahl von 400 Trainingsblöcken.

[0065] Die ersten zehn Eigenblöcke, die aus diesen Trainingsblöcken erzeugt werden, sind in Fig. 7 gezeigt.

Eigenblocksatz II

[0066] Ein zweiter Satz von Eigenblöcken wurde aus einem viel größeren Satz von Trainingsblöcken erzeugt. Diese Blöcke wurden aus 500 Gesichtsbildern im Trainingssatz genommen. In diesem Fall wurden die 16×16-Blöcke alle 8 Pixel abgetastet und überlappten sich somit um 8 Pixel. Dies erzeugte aus jedem 64×64-Trainingsbild 49 Blöcke und führte zu einer Gesamtzahl von 24.500 Trainingsblöcken.

[0067] Die ersten zwölf Eigenblöcke, die aus diesen Trainingsblöcken erzeugt wurden, sind in Fig. 8 dargestellt.

[0068] Empirische Ergebnisse zeigen, dass der Eigenblocksatz II etwas bessere Ergebnisse als der Satz I ergibt. Dies deshalb, weil er aus einem größeren Satz von aus Gesichtsbildern genommenen Trainingsblöcken berechnet wird und daher bei der Darstellung der Veränderungen in den Gesichtern als besser erkannt wird. Die Verbesserung des Leitungsvermögens ist jedoch nicht groß.

Aufbauen der Histogramme

[0069] Ein Histogramm wurde für jede abgetastete Blockposition in dem 64×64-Gesichtsbild aufgebaut. Die Anzahl der Histogramme hängt von dem Blockabstand ab. Zum Beispiel gibt es für einen Blockabstand von 16 Pixeln 16 mögliche Blockpositionen, und daher werden 16 Histogramme verwendet.

[0070] Der zum Aufbauen eines eine einzelne Blockposition darstellenden Histogramms verwendete Prozess ist in Fig. 9 gezeigt. Die Histogramme werden mittels eines großen Trainingssatzes **400** aus M Gesichtsbildern erzeugt. Für jedes Gesichtsbild weist der Prozess auf:

– Extrahieren **410** des relevanten Blocks aus einer Position (i,j) im Gesichtsbild.
– Berechnen der eigenblock-basierten Attribute für den Block und Bestimmen der relevanten Lagernummer **420** aus diesen Attributen.
– Erhöhen der relevanten Lagernummer im Histogramm **430**.

[0071] Dieser Prozess wird für alle M Bilder im Trainingssatz wiederholt, um ein Histogramm zu erzeugen, das eine gute Darstellung der Verteilung der Häufigkeit des Auftretens der Attribute ergibt. Idealerweise ist M sehr groß, z.B. einige tausend. Dies kann einfacher durch Verwenden eines Trainingssatzes erzielt werden, der aus einem Satz ursprünglicher Gesichter und mehreren hundert künstlichen Veränderungen jedes ursprünglichen Gesichts besteht.

Erzeugen der Histogrammlagernummer

[0072] Eine Histogrammlagernummer wird aus einem gegebenen Block mittels des folgenden Prozesses erzeugt, wie in Fig. 10 dargestellt. Der 16×16-Block **440** wird aus dem 64×64-Fenster oder Gesichtsbild extrahiert. Der Block wird auf den Satz **450** von A Eigenblöcken projiziert, um einen Satz von „Eigenblockgewichtungen" zu erzeugen. Diese Eigenblockgewichtungen sind die „Attribute", die in dieser Ausführungsform benutzt werden. Sie haben einen Bereich von –1 bis +1. Dieser Prozess ist in mehr Einzelheiten im Anhang B beschrieben. Jede Gewichtung wird in eine feste Anzahl von Niveaus L quantisiert, um einen Satz quantisierter Attribute **470** zu erzeugen, w_i, i = 1..A. Die quantisierten Gewichtungen werden wie folgt in einen einzelnen

Wert kombiniert:

$h = w_i L^{A-1} + w_2 L^{A-2} + w_3 L^{A-3} + ... + w_{A-1} L^1 + w_A L^0$,

wobei der erzeugte Wert h die Histogrammlagernummer **480** ist. Man beachte, dass die Gesamtzahl der Lager im Histogramm durch L^A gegeben ist.

[0073] Die „Lagerinhalte", d.h. die Häufigkeit des Auftretens des Satzes von Attributen, die zu dieser Lagernummer führen, können als ein Wahrscheinlichkeitswert angesehen werden, falls er durch die Anzahl von Trainingsbildern M geteilt wird. Weil jedoch die Wahrscheinlichkeiten mit einem Schwellenwert verglichen werden, gibt es tatsächlich keine Notwendigkeit, durch M zu teilen, da sich dieser Wert in den Berechnungen aufheben würde. Daher werden in den folgenden Erläuterungen die „Lagerinhalte" als „Wahrscheinlichkeitswerte" bezeichnet und so behandelt, als ob sie Wahrscheinlichkeitswerte sind, obwohl sie streng genommen tatsächlich Häufigkeiten des Auftretens sind.

[0074] Der obige Prozess wird sowohl in der Trainingsphase als auch in der Erkennungsphase verwendet.

Gesichtserkennungsphase

[0075] Der Gesichtserkennungsprozess beinhaltet ein Abtasten des Prüfbildes mit einem sich bewegenden 64×64-Fenster und ein Berechnen einer Gesichtswahrscheinlichkeit an jeder Fensterposition

[0076] Die Berechnung der Gesichtswahrscheinlichkeit ist in **Fig. 11** dargestellt. Für jede Blockposition im Fenster wird die Blocklagernummer **490** berechnet, wie im vorherigen Abschnitt beschrieben. Mittels des entsprechenden Histogramms **500** für die Position des Blocks wird jede Lagernummer abgerufen und die Wahrscheinlichkeit **510** dieser Lagernummer wird bestimmt. Die Summe **520** der Protokolle dieser Wahrscheinlichkeiten wird dann über alle Blöcke berechnet, um einen Gesichtswahrscheinlichkeitswert P_{face} (sonst als ein Protokollwahrscheinlichkeitswert bezeichnet) zu erzeugen.

[0077] Dieser Prozess erzeugt eine „Wahrscheinlichkeitsabbildung" für das gesamte Prüfbild. Mit anderen Worten wird ein Wahrscheinlichkeitswert bezüglich jeder möglichen Fenstermittenposition über das Bild abgeleitet. Die Kombination aller dieser Wahrscheinlichkeitswerte in eine rechtwinklig (oder irgendwie anders) geformte Anordnung wird dann als eine „Wahrscheinlichkeitsabbildung" entsprechend diesem Bild angesehen.

[0078] Die Abbildung wird dann invertiert, sodass der Prozess zum Finden eines Gesichts das Finden von Minimain der invertierten Abbildung beinhaltet (natürlich ist dies äquivalent zu einem Nicht-Invertieren der Abbildung und Auffinden der Maxima; eines von beiden kann durchgeführt werden). Eine so genannte abstands-basierte Technik wird verwendet. Diese Technik kann wie folgt zusammengefasst werden. Die Abbildungs-(Pixel-)Position mit dem kleinsten Wert in der invertierten Wahrscheinlichkeitsabbildung wird ausgewählt. Falls dieser Wert größer als ein Schwellenwert TD ist, werden keine weiteren Gesichter ausgewählt. Dies ist das Beendigungskriterium. Sonst wird ein gesichtsgroßer Block entsprechend der ausgewählten Mittenpixelposition gelöscht (d.h. in den folgenden Berechnungen nicht berücksichtigt) und die Gesichtspositionskandidaten-Findungsprozedur wird am Rest des Bildes wiederholt, bis das Beendigungskriterium erreicht ist.

Nichtgesichtsverfahren

[0079] Das Nichtgesichtsmodell weist einen zusätzlichen Satz von Histogrammen auf, die die Wahrscheinlichkeitsverteilung von Attributen in Nichtgesichtsbildern darstellen. Die Histogramme werden in exakt der gleichen Weise wie für das Gesichtsmodell erzeugt, außer dass die Trainingsbilder Beispiele von Nichtgesichtern anstelle von Gesichtern enthalten.

[0080] Während der Erkennung werden zwei Protokollwahrscheinlichkeitswerte berechnet, einer unter Verwendung des Gesichtsmodells und einer unter Verwendung des Nichtgesichtsmodells. Diese werden dann durch einfaches Subtrahieren der Nichtgesichtswahrscheinlichkeit von der Gesichtswahrscheinlichkeit kombiniert:

$p_{combined} = P_{face} - P_{nonface}$

[0081] $P_{combined}$ wird dann anstelle von P_{face} benutzt, um die Wahrscheinlichkeitsabbildung zu erzeugen (vor dem Invertieren).

[0082] Man beachte, dass der Grund, dass $P_{nonface}$ von P_{face} subtrahiert wird, darin liegt, dass diese Protokollwahrscheinlichkeitswerte sind.

[0083] Man beachte auch, dass die Gesichts- und die Nichtgesichtshistogramme optional am Ende des Trainingsprozesses (vor der Gesichtserkennung) durch einfaches Aufsummieren der Protokollhistogramme kombiniert werden können:

Summed histogram = log(histogram(face)) + log(histogram(non_face))

[0084] Dies ist der Grund, warum nur ein Histogramm für jede Blockposition/Haltung/Augenabstands – Kombination in der nachfolgenden Beschreibung erforderlich ist.

Histogrammbeispiele

[0085] Fig. 12a bis Fig. 12f zeigen einige Beispiele von Histogrammen, die durch den oben beschriebenen Trainingsprozess erzeugt wurden.

[0086] Fig. 12a, Fig. 12b und Fig. 12c sind von einem Trainingssatz von Gesichtsbildern abgeleitet, und Fig. 12d, Fig. 12e und Fig. 12f sind von einem Trainingssatz von Nichtsgesichtsbildern abgeleitet. Insbesondere:

	Gesichtshistogramme	Nichtgesichtshistogramme
gesamtes Histogramm	Figur 12a	Figur 12d
gezoomt auf die Hauptpeaks bei etwa h=1500	Figur 12b	Figur 12e
weiterer Zoom auf den Bereich um h=1570	Figur 12c	Figur 12f

[0087] Man kann klar erkennen, dass die Spitzen in dem Gesichtshistogramm und dem Nichtgesichtshistogramm an unterschiedlichen Stellen sind.

Histogrammspeicherung

[0088] Wie oben beschrieben, speichern die Histogramme statistische Informationen über die Wahrscheinlichkeit eines Gesichts bei einem gegebenen Maßstab und einer gegebenen Stelle in einem Bild. Die Reihenfolge der Histogramme ist jedoch unerwarteterweise für das Leistungsvermögen des Systems signifikant. Eine einfache Reihenfolge kann darin resultieren, dass der Zugriff nicht örtlich begrenzt ist (d.h. fortlaufende Zugriffe sind üblicherweise im Speicher weit entfernt). Dies kann eine schlechte Cache-Leistung bei der Realisierung mit Mikroprozessoren oder kundenspezifischen Prozessoren ergeben. Um dieses Problem zu behandeln, werden die Histogramme umgeordnet, sodass der Zugriff auf die Daten enger begrenzt wird.

[0089] Im vorliegenden Ausführungsbeispiel gibt es insgesamt sechs Histogramme:

F^{38}	Frontalgesicht mit einem Augenabstand von 38 Pixeln (d.h. ein „herangezoomtes“ Histogramm)
L^{38}	Gesicht um 25° nach links zeigend, mit einem Augenabstand von 38 Pixeln
R^{38}	Gesicht um 25° nach rechts zeigend, mit einem Augenabstand von 38 Pixeln

F^{22}	Frontalgesicht mit einem Augenabstand von 22 Pixeln (d.h. ein „Ganzgesichts“-Histogramm)
L^{22}	Gesicht um 25° nach links zeigend, mit einem Augenabstand von 22 Pixeln
R^{22}	Gesicht um 25° nach rechts zeigend, mit einem Augenabstand von 22 Pixeln

[0090] In der folgenden Erläuterung ist:
c der Wert von binmap (einer Abbildung, die den Histogrammeintrag für jede Stelle im Bild angibt) für eine gegebene Stelle in dem Bild bei einem gegebenen Maßstab – im vorliegenden Fall ist dies eine binäre 9-Bit-Zahl. Binmap wird durch Falten des Bildes mit neun Eigenblöcken, Quantisieren der resultierenden neun Eigenblockgewichtungen und ihr Kombinieren in einen einzelnen Wert vorberechnet;
x die x-Position in face window (zwischen 0 und 6); und
y die y-Position in face window (zwischen 0 und 6).

[0091] Dies bedeutet, dass die Histogramme für jede Pose (z.B. F^{38}) 512 × 7 × 7 = 25.088 Bytes groß sind.

[0092] $F^{38}_{c,x,y}$ ist der Wert des Histogramms für gegebene c, x und y.

[0093] Zum Beispiel ist $F^{38}_{15,4,5}$ der Wert, der durch das Frontalhistogramm mit dem Augenabstand von 38 an einer Position (4, 5) in face window für einen binmap-Wert von 15 gegeben ist.

[0094] Ein einfaches Ordnen der Histogramme im Speicher erfolgt nach c, dann x, dann y, dann pose und dann eye spacing. Ein schematisches Beispiel dieser Ordnung ist in Fig. 13a dargestellt. Ein verbessertes Ordnungssystem ist nach pose, dann x, dann y, dann c und dann eye spacing. Ein schematisches Beispiel dieser Art von Reihenfolge ist in Fig. 13b dargestellt.

[0095] Es gibt zwei Gründe für die Verbesserungen in der Cache-Leistung, wenn die Histogramme in der neuen Weise geordnet sind:
(i) die Weise, in der auf die Posen zugegriffen wird; und
(ii) die Weise, in der sich das Gesichtsfenster während einer Gesichtssuche bewegt.

[0096] Auf die zwei unterschiedlichen Posen (links, rechts und frontal) wird immer mit der gleichen Lagernummer und -stelle für jede Position zugegriffen. D.h., falls auf $F^{38}_{329,2,1}$ zugegriffen wird, dann wird auch auf $F^{38}_{329,2,1}$ und $R^{38}_{329,2,1}$ zugegriffen. Diese Werte sind im neuen Verfahren benachbart, sodass eine ausgezeichnete Cache-Leistung erzielt wird.

[0097] Das neue Verfahren zum Organisieren der Histogramme nimmt auch einen Vorteil der Weise, dass sich das Gesichtsfenster während einer Suche nach Gesichtern im Bild bewegt. Wegen des Weges, den sich das Gesichtsfenster bewegt, wird der gleiche c-Wert an vielen (x,y)-Positionen abgerufen.

[0098] Fig. 14 zeigt, welche Werte aus der Lagerabbildung benutzt werden, um ein Gesicht an einer bestimmten Stelle zu suchen. Zum Beispiel ist $F^{38}_{329,2,1}$ der Wert aus dem Frontalhistogramm für einen Augenabstand 38 für die (2,1)-Position im Gesichtsfenster.

[0099] Man kann sehen, dass, wenn sich das Gesichtserkennungsfenster um zwei Räume nach rechts bewegt, die hervorgehobenen Quadrate um eine Stelle nach links verschoben werden, d.h. der gleiche Wert wird an einer anderen Position abgerufen. In dem Beispiel von Fig. 14 wird $F^{38}_{329,2,1}$ zu $F^{38}_{329,1,1}$, wenn das Gesichtsfenster um zwei nach links verschoben worden ist.

[0100] Da der Algorithmus durch Schieben des Gesichtsfensters durch das Bild nach Gesichtern sucht, ruft er das gleiche binmap an mehreren Positionen ab. Dies bedeutet, dass, falls diese Werte im Speicher nahe zusammen gespeichert sind, dann die Cache-Leistung verbessert ist.

[0101] Eine weitere Verbesserung, die an der Histogrammstruktur entweder zusammen mit oder unabhängig von der oben beschriebenen Verbesserung vorgenommen werden kann, besteht darin, dass Seitenposen we-

niger Bits als Frontalposen benötigen. Die in jedem Histogrammlager gespeicherten Werte werden in Abhängigkeit davon, welche Pose sie darstellen, auf eine unterschiedliche Anzahl von Bits quantisiert.

[0102] Die Anzahl benutzter Bits für jedes der sechs Histogramme ist nachfolgend zusammengefasst:

F^{38}	Frontalgesicht mit einem Augenabstand von 38 Pixeln - 8 Bits
L^{38}	Gesicht um 25° nach links zeigend, mit einem Augenabstand von 38 Pixeln - 4 Bits
R^{38}	Gesicht um 25° nach rechts zeigend, mit einem Augenabstand von 38 Pixeln - 4 Bits
F^{22}	Frontalgesicht mit einem Augenabstand von 22 Pixeln - 8 Bits
L^{22}	Gesicht um 25° nach links zeigend, mit einem Augenabstand von 22 Pixeln - 4 Bits
R^{22}	Gesicht um 25° nach rechts zeigend, mit einem Augenabstand von 22 Pixeln - 4 Bits

[0103] Der Vorteil hiervon ist, dass jeder Satz von drei Histogrammwerten in 2 anstelle von 3 Bytes gespeichert werden kann.

[0104] Es wurde herausgefunden, dass dies möglich ist, weil die Seitenposen weniger Wichtigkeit als die Frontalpose für die Gesamtleistung des Algorithmus haben, und so können diese mit einer reduzierten Auflösung ohne signifikante Beeinflussung der Genauigkeit dargestellt werden.

Mehrmaßstabs-Gesichtserkennung

[0105] Um Gesichter unterschiedlicher Größen im Prüfbild zu erkennen, wird das Prüfbild durch einen Bereich von Faktoren skaliert, und eine Abstands- (d.h. Wahrscheinlichkeits-) Abbildung wird für jeden Maßstab erzeugt. In Fig. 15a bis Fig. 15c sind die Bilder und entsprechenden Abstandsabbildungen bei drei unterschiedlichen Maßstäben dargestellt. Das Verfahren gibt die beste Antwort (höchste Wahrscheinlichkeit oder minimaler Abstand) für den großen (zentralen) Gegenstand beim kleinsten Maßstab (Fig. 15a) und bessere Antworten für den kleineren Gegenstand (links der Hauptfigur) bei den größeren Maßstäben (eine dunklere Farbe auf der Abbildung stellt einen niedrigern Wert in der invertierten Abbildung dar, oder mit anderen Worten eine höhere Wahrscheinlichkeit, dass dort ein Gesicht ist). Kandidaten für Gesichtspositionen werden über unterschiedliche Maßstäbe extrahiert, indem zuerst die Position gefunden wird, die über alle Maßstäbe hinweg die beste Antwort gibt. Das heißt, die höchste Wahrscheinlichkeit (der niedrigste Abstand) wird aus allen Wahrscheinlichkeitsabbildungen bei allen Maßstäben ermittelt. Dieser Positionskandidat ist der erste, der als ein Gesicht gekennzeichnet wird. Das über diese Gesichtsposition zentrierte Fenster wird dann in jedem Maßstab aus der Wahrscheinlichkeitsabbildung gelöscht. Die Größe des gelöschten Fensters ist proportional zum Maßstab der Wahrscheinlichkeitsabbildung.

[0106] Beispiele dieses skalierten Löschvorgangs sind in Fig. 15a bis Fig. 15c dargestellt. Insbesondere wird die höchste Wahrscheinlichkeit über alle Abbildungen auf der linken Seite der Abbildung mit größtem Maßstab (Fig. 15c) gefunden. Ein Bereich **530** entsprechend der vermuteten Größe eines Gesichts wird in Fig. 15c gelöscht. Dementsprechend, aber skaliert, werden Bereiche **532**, **534** in den kleineren Abbildungen gelöscht.

[0107] Bereiche größer als das Prüffenster können in den Abbildungen gelöscht werden, um überlappende Erkennungen zu vermeiden. Insbesondere ist ein Bereich gleich der Größe des Prüffensters, der von einer Grenzlinie der halben Breite/Länge wie das Prüffenster umgeben ist, geeignet, um solche überlappenden Erkennungen zu vermeiden.

[0108] Weitere Gesichter werden durch Suchen nach der nächstbesten Antwort und Löschen der entsprechenden Fenster nacheinander erkannt.

[0109] Die zwischen den bearbeiteten Maßstäben erlaubten Abstände werden durch die Empfindlichkeit des Verfahrens auf Veränderungen in der Größe beeinflusst. Es wurde in dieser vorläufigen Studie der Maßstabsinvarianz herausgefunden, dass das Verfahren nicht übermäßig empfindlich auf Veränderungen der Größe ist, da Gesichter, die in einem bestimmten Maßstab eine gute Antwort geben, häufig auch bei benachbarten Maßstäben eine gute Antwort gaben.

[0110] Die obige Beschreibung bezieht sich auf das Erkennen eines Gesichts, obwohl die Größe des Gesichts im Bild zu Beginn des Erkennungsvorgangs nicht bekannt ist. Ein weiterer Aspekt der Mehrmaßstabs-Gesichtserkennung ist die Verwendung zwei oder mehr parallelen Erkennungen in unterschiedlichen Maßstäben, um den Erkennungsvorgang zu bestätigen. Dies kann Vorteile haben, falls z.B. das zu erkennende Gesicht teilweise dunkel ist, oder die Person einen Hut trägt, usw..

[0111] Fig. 15d bis Fig. 15g zeigen schematisch diesen Prozess. Während der Trainingsphase wird das System an Fenstern (in jeweilige Blöcke geteilt, wie oben beschrieben), die das gesamte Prüfgesicht angeben, trainiert (Fig. 15d), um „Ganzgesichts"-Histogrammdaten zu erzeugen, und auch Fenstern in einem erweiterten Maßstab, sodass nur ein zentraler Bereich des Prüfgesichts eingeschlossen ist (Fig. 15e), um „herangezoomte" Histogrammdaten zu erzeugen. Dies erzeugt zwei Sätze von Histogrammdaten. Ein Satz betrifft die „Ganzgesichtsfenster" von Fig. 15d, und der andere betrifft die „Zentralgesichtsbereichsfenster" von Fig. 15e.

[0112] Während der Erkennungsphase wird für ein gegebenes Prüffenster **536** das Fenster auf zwei unterschiedliche Maßstäbe des Prüfbildes angewendet, sodass in einem (Fig. 15f) das Prüffenster die gesamte erwartete Größe eines Gesichts umgibt und im anderen (Fig. 15g) das Prüffenster den zentralen Bereich eines Gesichts in dieser erwarteten Größe einschließt. Diese werden jeweils wie oben beschrieben verarbeitet, wobei sie mit den jeweiligen Sätzen von Histogrammdaten entsprechend der Art des Fensters verglichen werden. Die Protokollwahrscheinlichkeiten aus jedem parallelen Prozess werden addiert, bevor der Vergleich mit einem Schwellenwert angewendet wird.

[0113] Das Zusammenführen dieser beiden Aspekte der Mehrmaßstabs-Gesichtserkennung führt zu einer besonders eleganten Einsparung der Datenmenge, die gespeichert werden muss.

[0114] Insbesondere sind in diesen Ausführungsbeispielen die mehreren Maßstäbe für die Anordnungen der Fig. 15a bis Fig. 15c in einer geometrischen Folge angeordnet. Im vorliegenden Ausführungsbeispiel ist jeder Maßstab in der Folge einen Faktor von unterschiedlich zum benachbarten Maßstab in der Folge. Dann wird für die unter Bezug auf Fig. 15d bis Fig. 15g beschriebene parallele Erkennung die Erkennung des größeren Maßstabs und des mittleren Bereichs mit einem Maßstab um drei Stufen höher in der Folge, d.h. $2^{3/4}$-fach größer als der Maßstab des „vollen Gesichts" unter Verwendung von Attributdaten betreffend den Maßstab drei Stufen höher in der Folge ausgeführt. Somit bedeutet der geometrische Fortschritt außer an den Extremen des Bereichs der mittleren Maßstäbe, dass die parallele Erkennung der Fig. 15d bis Fig. 15g immer unter Verwendung von Attributdaten ausgeführt werden kann, die bezüglich eines weiteren Maßstabs der Stufen höher in der Folge erzeugt werden.

[0115] Die zwei Prozesse (Mehrmaßstabserkennung und Parallelmaßstabserkennung) können auf verschiedene Weise kombiniert werden. Zum Beispiel kann der Mehrmaßstabserkennungsprozess von Fig. 15a bis Fig. 15c zuerst angewendet werden und dann kann der Parallelmaßstabserkennungsprozess von Fig. 15d bis Fig. 15g auf Bereiche (und Maßstäbe) angewendet werden, die während des Mehrmaßstabserkennungsprozesses identifiziert wurden. Jedoch kann ein bequemer und effizienter Gebrauch der Attributdaten erreicht werden durch:

– Ableiten von Attributen bezüglich des Prüffensters in jedem Maßstab (wie in Fig. 15a bis Fig. 15c)
– Vergleichen dieser Attribute mit „Vollgesichts"-Histogrammdaten, um einen „Vollgesichtssatz" von Abstandsabbildungen zu erzeugen
– Vergleichen der Attribute mit den „herangezoomten" Histogrammdaten, um einen „herangezoomten" Satz von Abstandsabbildungen zu erzeugen
– für jeden Maßstab n: Kombinieren der „Vollgesichts"-Abstandsabbildung für den Maßstab n mit der „herangezoomten" Abstandsabbildung für den Maßstab n + 3
– Ableiten von Gesichtspositionen aus den kombinierten Abstandsabbildungen, wie oben unter Bezug auf Fig. 15a bis Fig. 15c beschrieben.

[0116] Ferner kann eine Parallelprüfung durchgeführt werden, um unterschiedliche Posen zu erkennen, wie beispielsweise Schauen geradeaus, Schauen teilweise nach oben, nach unten, nach links, nach rechts, usw.. Hierbei ist ein jeweiliger Satz von Histogrammdaten erforderlich, und die Ergebnisse werden vorzugsweise mit-

tels einer „max"-Funktion kombiniert, d.h. die die höchste Wahrscheinlichkeit ergebende Pose wird weiter zur Schwellenwertberechnung geführt, die anderen werden verworfen.

Verbesserte Verwendung mehrerer Maßstäbe

[0117] Der Erkennungsalgorithmus sieht viele Wahrscheinlichkeitsabbildungen in vielen Maßstäben vor. Das Erfordernis ist, alle Stellen im Bild zu finden, wo die Wahrscheinlichkeit einen gegebenen Schwellenwert übersteigt, während gleichzeitig sichergestellt wird, dass es keine überlappenden Gesichter gibt.

[0118] Ein Nachteil des oben beschriebenen Verfahrens ist, dass es die Speicherung eines kompletten Satzes von Wahrscheinlichkeitsabbildungen in allen Maßstäben erfordert, was eine Anforderung eines großen Speichers ist. Die folgende Technik erfordert nicht die Speicherung aller Wahrscheinlichkeitsabbildungen gleichzeitig.

[0119] Zusammenfassend wird eine vorübergehende Liste von Gesichtspositionskandidaten gehalten. Wenn die Wahrscheinlichkeitsabbildung für jeden Maßstab berechnet ist, werden die Wahrscheinlichkeitsmaxima gefunden und mit der Liste der Gesichtspositionskandidaten verglichen, um zu gewährleisten, dass keine überlappenden Gesichter existieren.

[0120] Im Detail benutzt dieses Verfahren eine face list, um eine Liste aktueller Positionen zu halten, wenn es ein Gesicht geben kann. Jedes Gesicht in der face list hat eine face location und eine face size. Der treshold ist der Wahrscheinlichkeitsschwellenwert, oberhalb dessen ein Objekt als ein Gesicht angenommen wird. Der scale factor ist der Größenfaktor zwischen aufeinanderfolgenden Maßstäben (1,189207116 der $\sqrt[4]{2}$ im vorliegenden Ausführungsbeispiel).

[0121] In der folgenden beispielhaften Beschreibung wird eine face size von 16 × 16 betrachtet.

[0122] Der Prozess ist schematisch im Flussdiagramm von Fig. 16 veranschaulicht.

[0123] Bezug nehmend auf Fig. 16 startet der Prozess in einem Schritt **1400** bezüglich eines der Maßstäbe (im gezeigten Beispiel der kleinste Maßstab). Das erste Mal, dass der Schritt **1400** stattfindet, wird face list leer sein, aber im Allgemeinen wird für alle Gesichter in der face list die Gesichtsgröße für jedes Gesicht in Schritt **1400** durch Multiplizieren der jeweiligen Gesichtsgröße mit dem scale factor modifiziert. Dies stellt sicher, dass die erkannten Gesichter bezüglich des vorherigen Maßstabs für einen gültigen Vergleich mit Maxima im aktuellen Maßstab korrekt dimensioniert sind.

[0124] In einem Schritt **1410** wird der maximale Wahrscheinlichkeitswert mp in der aktuellen Abbildung erfasst.

[0125] In einem Schritt **1420** wird der maximale Wahrscheinlichkeitswert mp mit dem Schwellenwert threshold verglichen. Falls mp größer als threshold ist, dann geht die Steuerung weiter zu einem Schritt **1430**. Falls dagegen mp nicht größer als threshold ist, dann wird eine Verarbeitung der nächsten Abbildung (entsprechend dem nächsten zu behandelnden Maßstabsfaktor) in einem Schritt **1440** begonnen.

[0126] Zurück zu Schritt **1430** geht, falls die Stelle in der Wahrscheinlichkeitsabbildung des aktuellen Maßstabes für den maximalen Wert mp mit einem Gesicht in der face list (die in Schritt **1400** abgeleiteten modifizierten Größen betrachtend) überlappt (zusammenfällt), dann die Steuerung weiter zu einem Schritt **1450**. Falls nicht, geht die Steuerung zu einem Schritt **1460**.

[0127] In Schritt **1450** wird der Wert mp mit einem gespeicherten Wahrscheinlichkeitswert bezüglich des existierenden Gesichts verglichen. Falls mp größer als diese Wahrscheinlichkeit ist, dann wird das existierende Gesicht in einem Schritt **1470** gelöscht und ein neuer Eintrag in der face list entsprechend dem aktuellen Wert und der aktuellen Position von mp erzeugt. Insbesondere wird der Wert mp bezüglich des neuen Eintrags in der face list gespeichert und ein um die Bildposition der aktuellen maximalen Wahrscheinlichkeit zentrierter 16×16-Pixelbereich wird in einem Schritt **1480** auf den threshold gesetzt. In einem Schritt **1490** wird die aktuelle Position des maximalen Wahrscheinlichkeitswerts der face list mit einer Gesichtsgröße von 16 hinzugefügt. Die Steuerung kehrt dann zu Schritt **1410** zurück.

[0128] Zurück zu **1460** wird, falls die maximale Wahrscheinlichkeitsposition als nicht mit irgendeinem Gesicht in der face list überlappend erkannt wurde (in Schritt **1430**), dann ein neuer Eintrag in der face list erzeugt. Wie

oben erläutert, wird in Schritt **1460** der Wert mp gespeichert und ein 16×16 Bereich um den aktuellen Maximalwert wird auf den threshold gesetzt. In einem Schritt **1465** wird die aktuelle maximale Position der face list mit einer Gesichtsgröße von 16 hinzugefügt und die Steuerung kehrt zu Schritt **1410** zurück.

[0129] Falls in Schritt **1450** der maximale Wahrscheinlichkeitswert mp als nicht größer als die Wahrscheinlichkeit des existierenden (überlappenden) Gesichts erfasst wird, dann geht die Steuerung zu einem Schritt **1455**, in dem der Bereich des existierenden Gesichts auf den Wert threshold gesetzt wird, und die Steuerung kehrt zu Schritt **1410** zurück.

[0130] In jeder dieser Stufen wird, wenn die Steuerung zu Schritt **1410** zurückkehrt, ein maximaler Wahrscheinlichkeitswert mp wiedererkannt, aber dies erfolgt im Licht der Modifikationen an den Wahrscheinlichkeitswerten um die erkannten Gesichter in den Schritten **1460**, **1455** und **1480**. Somit laufen die in diesen Schritten erzeugten modifizierten Werte tatsächlich nicht durch den Test von Schritt **1420**, in dem ein gleich dem Wert threshold gesetzter Wert als ihn nicht übersteigend gefunden wird. Demgemäß wird Schritt **1420** ermitteln, ob eine weitere Position in der aktuellen Abbildung existiert, wo der Schwellenwert überschritten wird.

[0131] Ein Vorteil dieses Verfahren ist, dass es die separate Betrachtung jedes Maßstabs der Wahrscheinlichkeitsabbildung erlaubt. Nur die face list muss zwischen der Verarbeitung jedes Maßstabs gespeichert werden. Dies hat die folgenden Vorteile:

– Geringere Speicheranforderung: Es muss kein kompletter Satz von Wahrscheinlichkeitsabbildungen gespeichert werden. Nur die Gesichtsliste muss gespeichert werden, was viel weniger Speicher erfordert.

– Erlaubt Temporaldezimierung: Der Algorithmus kann Verfahren, wie beispielsweise eine Temporaldezimierung verwenden, bei welcher eine Verarbeitung für ein Vollbild zwischen mehreren Zeitschlitzen aufgeteilt wird und nur ein Untersatz von Maßstäben während jedes Zeitschlitzes verarbeitet wird. Dieses Verfahren kann nun benutzt werden, wobei zwischen jedem Aufruf anstelle des gesamten Satzes von berechneten Wahrscheinlichkeitsabbildungen bisher nur eine Gesichtsliste gehalten werden muss.

– Erlaubt schnellere Suche: Nur ein Maßstab wird zu einer Zeit betrachtet. Deshalb müssen nicht jedes Mal, wenn ein Maximum gefunden wird, Bereiche über alle Maßstäbe in einem Satz von Wahrscheinlichkeitsabbildungen gelöscht werden.

Änderungserfassung

[0132] In Situationen, wo die Gesichteserkennung in Echtzeit ausgeführt werden muss, kann es schwierig sein, die gesamte Gesichtserkennungsverarbeitung in der zulässigen Zeit – z.B. eine Vollbildperiode eines Videosignals – abzuschließen.

[0133] Es wird ein Änderungsverfassungsprozess verwendet, um zu erkennen, welche Bereiche des Bildes seit dem vorherigen Vollbild geändert worden sind, oder wenigstens bestimmte Bereiche, die als seit dem vorherigen Vollbild nicht verändert erkannt werden, aus dem Gesichtserkennungsprozess zu entfernen.

[0134] Bereiche des Bildes, die seit dem vorherigen Vollbild nicht geändert worden sind, müssen an sich nicht wieder eine Gesichtserkennung durchführen lassen, da das Ergebnis wahrscheinlich gleich dem vorherigen Vollbild ist. Bereiche des Bildes, die geändert worden sind, müssen jedoch an sich eine neue Gesichtserkennung durchführen lassen. Diese Bereiche des Bildes werden während der Änderungserkennung als „interessierende Bereiche" gekennzeichnet.

[0135] Im vorliegenden Ausführungsbeispiel wird eine Änderungserkennung nur in einem einzelnen festen Maßstab (z.B. dem ursprünglichen Bildmaßstab) oder dem größten Maßstab, der in der Gesichtserkennung erfasst wird, durchgeführt. Der Prozess ist in **Fig. 17** veranschaulicht, die schematisch einen Bewegungsdetektor zeigt.

[0136] Die aktuellen und die vorherigen Vollbilder werden zuerst durch Tiefpassfilter **1100**, **1110** verarbeitet. Die zwei Vollbilder werden dann einem Differenzbilder **1120** zugeführt, um ein Vollbild-Differenzbild zu erzeugen, zum Beispiel eine Darstellung der absoluten Pixel- (oder Block-) Unterschiede zwischen Vollbildern mit einem Differenzwert je Pixel- (oder Block-) Position. Die absoluten Werte des Differenzbildes werden dann durch einen Vergleich mit einem Schwellenwert Thr_{diff} einer Schwellenwertberechnung **1130** unterzogen, um ein binäres Differenzbild zu erzeugen, d.h. eine Anordnung von Ein-Bit-Werten mit einem Wert je Pixel- (oder Block-) Position: sehr kleine Unterschiede werden auf Null gesetzt (keine Veränderung) und größere Unterschiede werden auf Eins gesetzt (Änderung erkannt). Schließlich wird ein morphologischer Öffnungsvorgang **1140** an dem binären Differenzbild durchgeführt, um sequenziellere Bereiche erkannter Veränderung/Bewe-

gung zu erzeugen.

[0137] In der Praxis kann auf den Tiefpassfiltervorgang verzichtet werden.

[0138] Das morphologische Öffnen ist eine bekannte Bildverarbeitungstechnik und wird in diesem Beispiel an einem 3×3-Bereich durchgeführt (d.h. ein 3×3-Block wird als morphologisches Strukturelement benutzt) und weist einen morphologischen Erosionsvorgang auf, gefolgt von einem morphologischen Ausdehnungsvorgang. Um dies daran auszuführen, was grundsätzlich ein rasterbasiertes System ist, wird die morphologische Verarbeitung nach einem Verarbeiten jeder dritten Zeile ausgeführt.

[0139] Die Änderungserfassung kann auf das gesamte Bild angewendet werden, wie oben beschrieben, um eine Abbildung von Bereichen des Bildes zu erzeugen, wo Änderungen erkannt worden sind. Die Gesichtserkennung wird auf diese Bereiche angewendet.

[0140] Alternativ kann die Änderungserfassung benutzt werden, um bestimmte Bereiche des Bildes von der Gesichteserkennung auszuschließen, aber ohne notwendiges Erkennen aller Bereiche von Bewegung oder „keine Bewegung". Diese Technik hat den Vorteil des Verringerns der Verarbeitungsanforderungen des Änderungserfassungsprozesses, während nach wie vor potenziell ein nützliches Einsparen bei der Verarbeitung für die Gesichtserkennung selbst bereitgestellt wird. Ein schematisches Beispiel dieses Prozesses ist in Fig. 18a bis Fig. 18c veranschaulicht.

[0141] In Fig. 18a wird die Änderungserfassung in einer Rasterabtastanordnung angewendet, bei welcher ein Scan **1150** von horizontalen Zeilen (von Pixeln oder Blöcken) von oben links nach unten rechts eines Bildes durchgeführt wird. Der in Fig. 17 dargestellte Basisprozess (ohne morphologische Verarbeitung und vorzugsweise ohne Tiefpassfiltern) wird verwendet und das Bild wird mit dem vorherigen Bild verglichen. An jedem Abtastpunkt wird die erfasste Absolutdifferenz mit dem Schwellenwert Thr_{diff} verglichen.

[0142] Der Scan **1150** schreitet fort, bis die erfasste Absolutdifferenz bezüglich einer Abtastposition **1160** den Schwellenwert Thr_{diff} übersteigt. An diesem Punkt endet der Scan **1150**.

[0143] Drei ähnliche Scans **1170**, **1180**, **1190** werden ausgeführt. Der Scan **1170** ist ein horizontaler Scan, beginnend am unteren Rand des Bildes, und endet, wenn eine Abtastposition **1200** zu einem den Schwellenwert Thr_{diff} übersteigenden Absolutdifferenzwert führt. Der Scan **1180** ist ein vertikaler Abwärtsscan, beginnend auf der linken Seite des Bildes, und endet, wenn eine Abtastposition **1210** zu einem den Schwellenwert Thr_{diff} übersteigenden Absolutdifferenzwert führt. Und der Scan **1190** ist ein vertikaler Abwärtsscan, beginnend auf der rechten Seite des Bildes, und endet, wenn eine Abtastposition **1220** zu einem den Schwellenwert Thr_{diff} übersteigenden Absolutdifferenzwert führt.

[0144] Die vier Punkte **1160**, **1200**, **1210**, **1220** definieren einen Bereichskasten **1230** in Fig. 18e. Insbesondere sind, falls die Bildkoordinaten eines Punktes nnnn (x_{nnnn}, y_{nnnn}) sind, dann die vier Scheitelpunkte des Bereichskastens **1230** gegeben durch:

oben links	(X_{1210}, y_{1160})
oben rechts	(X_{1220}, y_{1160})
unten links	(X_{1210}, y_{1200})
unten rechts	(X_{1220}, y_{1200})

[0145] Die Bereichsbox definiert deshalb nicht alle Bereiche des Bildes, in denen Änderungen erfasst worden sind, sondern sie definiert einen Bereich (außerhalb der Bereichsbox), der von der Gesichtsverarbeitung ausgeschlossen werden kann, weil dort keine Veränderung erfasst worden ist. Hinsichtlich des Bereichs innerhalb des Bereichskastens wurde potenziell der gesamte Bereich verändert, aber es ist eine üblichere Situation, dass einige Teile dieses Bereichs verändert worden sind und einige nicht.

[0146] Natürlich gibt es einige Permutationen dieser Techniken:
(a) die Reihenfolge, in denen die vier Suchen durchgeführt werden
(b) die Richtung, in welche jede Suche durchgeführt wird (die Pfeile können in jedem Diagramm umgekehrt werden, ohne die Wirkung des Algorithmus zu verändern)
(c) die Scans können entweder sequenziell (ein Scan nach dem anderen) oder parallel (zwei oder mehr

Scans gleichzeitig) ausgeführt werden.

[0147] In einer schematisch in Fig. 19a bis Fig. 19c dargestellten Abwandlung werden die zwei vertikalen Scans **1180'**, **1190'** nur bezüglich jener Spalten **1240** ausgeführt, die durch die zwei horizontalen Scans **1150**, **1170** noch nicht ausgeschlossen worden sind. Diese Abwandlung kann die Verarbeitungsanforderungen reduzieren.

[0148] Die oben beschriebenen Änderungserfassungstechniken arbeiten mit den Gesichtserkennungstechniken gut wie folgt. Die Änderungserfassung wird ausgeführt, beginnend an vier Extrempunkten (Ecken) des Bildes und endet in jedem Fall, wenn eine Veränderung erfasst wird. Somit wird außer dem potenziell letzten Pixel (oder Block) oder der Teilspalte jedes der Änderungserfassungsprozesse die Änderungserfassung nur bezüglich jener Bildbereiche ausgeführt, die keiner Gesichtserkennung unterzogen werden. Analog wird außer diesem dieser letzten Pixel, Block oder Teilspalte die Gesichtserkennung nur bezüglich Bereichen ausgeführt, die keinem Änderungserfassungsprozess unterzogen worden sind. Unter Berücksichtigung, dass die Änderungserfassung weniger prozessorintensiv ist als die Gesichtserkennung, bedeutet diese relativ winzige Überlappung zwischen den zwei Prozessen, dass in beinahe allen Situationen die Verwendung der Änderungserfassung die gesamten Verarbeitungsanforderungen eines Bildes reduzieren wird.

[0149] Ein anderes Verfahren der Änderungserfassung wendet bewegungscodierte Signale, wie beispielsweise MPEG-codierte Signale, oder solche, die im Voraus in dieser Form codiert und zur Gesichtserkennung decodiert worden sind, an. Bewegungsvektoren oder dergleichen zu den Signalen können angeben, ob eine Zwischenbildveränderung stattgefunden hat. Ein Block (z.B. ein MPEG-Makroblock) am Ziel (in einem aktuellen Bild) jedes Bewegungsvektors kann als ein Bereich der Veränderung markiert werden. Dies kann anstelle oder zusätzlich zu den oben beschriebenen Änderungserfassungserfassungstechniken erfolgen.

[0150] Ein weiteres Verfahren zum Reduzieren der Prozessanforderungen ist wie folgt. Der Gesichtserkennungsalgorithmus wird in eine Anzahl von Stufen geteilt, die über viele Maßstäbe wiederholt werden. Der Algorithmus wird erst nach n Aufrufen beendet. Der Algorithmus wird automatisch so unterteilt, dass jeder Aufruf etwa ein gleiches Zeitmaß benötigt. Die Schlüsselmerkmale dieses Verfahrens sind:

– Das Verfahren verwendet ein automatisches Verfahren zum Unterteilen des Algorithmus in Stücke, die ein gleiches Maß Verarbeitungszeit benötigen.
– Für die von jeder Stufe benötigte Verarbeitungszeit werden Schätzungen verwendet, sodass der Algorithmus abbrechen kann, bevor eine gegebene Stufe ausgeführt wird, falls sie zuviel Zeit benötigen wird.
– Der Algorithmus kann erst am Ende jeder Stufe abbrechen; er kann nicht mitten in einer Stufe abbrechen. Dies begrenzt die Menge des erforderlichen lokalen Speichers und vereinfacht die Programmablaufsteuerung.
– Die Schätzungen für die für jede Stufe benötigte Verarbeitungszeit können automatisch aus echt gemessenen Zeitinformationen verfeinert werden – obwohl dies derzeit nicht erfolgt.
– Eine Nachführung von Gesichtern kann am Ende jedes Aufrufs erfolgen, sodass selbst ohne eine neue Gesichtserkennung, um eine neue Gesichtsposition zu ermitteln, die Hautfarbennachführung dem Gesicht fortlaufend folgt.
– Alternativ kann das Nachführen erst im (n + 1)-ten Aufruf durchgeführt werden (nachdem alle n Gesichtserkennungsaufrufe abgeschlossen worden sind). Dies bedeutet, dass die Nachführung mit einer geringeren Rate durchgeführt wird und damit weniger genau ist, aber es erlaubt ein Verkleinern des Algorithmus auf eine kürzere Ausführungszeit je Aufruf, falls dies erforderlich ist.

[0151] Die folgende Tabelle zeigt die Reihenfolge, in der die Stufen für eine Gesichtssuche über sechs Maßstäbe ausgeführt werden:

Prozess	Maßstab	Beschreibung
Bewegung	1	Bewegung wird benutzt, um den Suchbereich zu verringern
Variation	1	Bildvariation wird benutzt, um den Suchbereich zu verringern

Dezimieren	1 → 2	Bild wird auf den nächsten Maßstab verkleinert
Falten	2	Bild wird gefaltet, um eine Lagerabbildung zu erzeugen
Dezimieren	2 → 3	
Falten	3	
Dezimieren	3 → 4	
Falten	4	
Nachschlagen	4	Lagerabbildungen werden benutzt, um Gesichtswahrscheinlichkeiten nachzuschlagen
Maxima-Suche	4	Maximumwahrscheinlichkeiten werden gefunden und einer Schwellenwertberechnung unterzogen
Dezimieren	4 → 5	
Falten	5	
Nachschlagen	5	
Maxima-Suche	5	
Dezimieren	5 → 6	
Falten	6	
Nachschlagen	6	
Maxima-Suche	6	
Nachführen	- -	

[0152] Die folgende Tabelle zeigt, was passieren kann, falls eine Dezimaldezimierung von 4 verwendet wird. Der Algorithmus teilt die Verarbeitung automatisch in Blöcke gleicher Zeit – dies wird durch die Tatsache verkompliziert, dass die Verarbeitung für die früheren Maßstäbe mehr Zeit erfordert als die Verarbeitung für spätere Maßstäbe (da die Bilder für die früheren Maßstäbe größer sind). Der Algorithmus schätzt das Zeitmaß, das jede Stufe benötigt, bevor sie ausgeführt wird. Diese Schätzung ergibt sich aus dem speziellen Prozess und der Anzahl zu verarbeitender Pixel für einen gegebenen Maßstab. Zum Beispiel könnten die folgenden Schätzungen verwendet worden sein:

Bewegung:	1 je Pixel
Variation:	1 je Pixel
Dezimieren:	2 je Pixel
Falten:	3 je Pixel
Nachschlagen:	4 je Pixel
Maxima-Suche:	3 je Pixel

Prozess	Maßstab	Aufruf	Anzahl Pixel im Maßstab	erforderliche Verarbeitungs-einheiten	kumulative Verarbeitungs-einheiten je Aufruf
Bewegung	1	1	1000	1000	1000
Variation	1	1	1000	1000	2000
Dezimieren	1 → 2	1	1000	2000	4000
Nachführen	- -	1			
		zurück			
Falten	2	2	700	2100	2100
Dezimieren	2 → 3	2	700	1400	3500
Falten	3	2	500	1500	5000
Nachführen	- -	2			
		zurück			
Dezimieren	3 → 4	3	500	1000	1000
Falten	4	3	354	1062	2062
Nachschlagen	4	3	354	1416	3478
Maxima-Suche	4	3	354	1062	4540
Dezimieren	4 → 5	3	354	708	5248
Nachführen	- -	3			
		zurück			
Falten	5	4	250	750	750
Nachschlagen	5	4	250	1000	1750
Maxima-Suche	5	4	250	750	2500
Dezimieren	5 → 6	4	177	354	2854
Falten	6	4	177	531	3385
Nachschlagen	6	4	177	708	4093
Maxima-Suche	6	4	177	531	4624
Nachführen		4			
		zurück			

[0153] Es ist möglich, im Voraus zu berechnen, dass die gesamte Verarbeitung für alle Maßstäbe 18.872 Verarbeitungseinheiten benötigt. Dies bedeutet, dass zum Aufteilen der Verarbeitung in vier gleiche Blöcke das System etwa 4.718 Verarbeitungseinheiten je Aufruf ausführen muss.

[0154] Bevor jede Stufe ausgeführt wird, berechnet das System, ob die Anzahl erforderlicher kumulativer Verarbeitungseinheiten über ein vorbestimmtes Niveau (z.B. 5.300) steigen wird. Falls dies der Fall ist, dann wird abgebrochen, ohne diese Stufe auszuführen. Dies hat gegenüber der Zeitgebung den Vorteil, dass vor einer Durchführung bekannt ist, ob sie die zugeordnete Zeit übersteigen wird.

Raumdezimierung

[0155] Raumdezimierung ist eine weitere Technik, die verwendet werden kann, um die Verarbeitungsanforderungen bezüglich jedes Bildes zu reduzieren.

[0156] Es werden zwei Beispiele von Raumdezimierung beschrieben: Ein Verfahren durchsucht das Bild in horizontalen Streifen, das andere durchsucht das Bild in einer ausgedünnten Weise.

[0157] Das horizontale Durchstreifen wurde gewählt, weil es als effizienter angesehen wird, horizontale Streifen zu verarbeiten, obwohl irgendeine Art Aufteilung verwendet werden könnte (Quadrate, vertikale Streifen, usw.).

[0158] Es ist auch möglich, die Raumdezimierung in einer ausgedünnten Weise zu steuern. Dies ist nicht so effizient wie das Verwenden von Streifen, aber es kann bessere Ergebnisse bringen, da Gesichter bei jedem Aufruf des Algorithmus noch irgendwo im gesamten Bild erkannt werden können.

[0159] Fig. 20a bis Fig. 20c zeigen schematisch eine Streifentechnik. In Fig. 20a wird keine Raumdezimierung verwendet und das gesamte Bild **1300** wird einer Gesichtserkennung unterzogen. In Fig. 20b wird das Bild in zwei Teile **1310**, **1320** geteilt. Diese werden einer Gesichtserkennung in abwechselnden Bildern unterzogen. In Fig. 20c wird das Bild in drei Teile **1330**, **1340**, **1350** geteilt, sodass jeder Teil einer Gesichtserkennung einzeln in allen drei Bildern unterzogen wird. Die Teile können unterschiedlich sein oder können etwas überlappen.

[0160] Fig. 21a bis Fig. 21d zeigen schematisch eine so genannte ausgedünnte Raumdezimierung.

[0161] Es sind drei Variablen definiert:
SparseX: ist ein Dezimierungsgrad in der horizontalen Richtung. Falls
SparseX=1, dann gibt es keine horizontale Dezimierung. Falls SparseX=2, dann werden abwechselnde Pixel oder Blöcke einer Gesichtserkennung in jedem Bild unterzogen, sodass jede Pixel- oder Blockposition alle zwei Bilder einer Gesichtserkennung unterzogen wird usw..
SparseY: ist das äquivalente Dezimierungsmaß der vertikalen Richtung.
UseChequerBoard: ist auf Null gesetzt, falls die Dezimierung in aufeinanderfolgenden Spalten ausgerichtet ist, und auf Eins, falls sie zwischen zwei aufeinanderfolgenden Reihen oder Spalten (oder Gruppen von Reihen/Spalten) versetzt ist.

[0162] Die Beispiele von Fig. 21a bis Fig. 21d sind wie folgt:

	Fig. 21a	Fig. 21b	Fig. 21c	Fig. 21d
SparseX	2	2	2	2
SparseY	1	1	2	2
UserChequerBoard	0	1	0	1

[0163] Eine Kombination von räumlicher und zeitlicher Dezimierung kann verwendet werden. Zum Beispiel könnten diskrete Raumabschnitte eines Bildes (z.B. ein Drittel eines Bildes) über eine Anzahl von Vollbildern verarbeitet werden. Die auf diese Weise verarbeiteten Teile (z.B. die drei Drittel) könnten aus dem gleichen Bild oder aus unterschiedlichen Bildern kommen.

Gesichtsnachführung

[0164] Es wird nun ein Gesichtsnachführungsalgorithmus beschrieben. Der Nachführungsalgorithmus soll die Gesichtserkennungsleistung in Bildfolgen verbessern.

[0165] Der Ausgangspunkt des Nachführungsalgorithmus ist das Erkennen jeden Gesichts in jedem Vollbild einer Bildfolge. Es wird jedoch erkannt, dass ein Gesicht in der Folge manchmal nicht erkannt werden kann. Unter diesen Umständen kann der Nachführungsalgorithmus ein Interpolieren über die fehlenden Gesichtserkennungen unterstützen.

[0166] Letztlich ist das Ergebnis der Gesichtsnachführung in der Lage, brauchbare Metadaten aus jedem

Satz Vollbilder, die zur gleichen Szene in einer Bildfolge gehören, auszugeben. Dies können enthalten:
– Anzahl von Gesichtern.
– „Verbrecherfoto" (Umgangswort für ein Bild eines Gesichts einer Person, abgeleitet aus einem Begriff betreffend ein Polizeidateifoto) jedes Gesichts.
– Vollbildnummer, bei welcher das jeweilige Gesicht zuerst erscheint.
– Vollbildnummer, bei welcher das jeweilige Gesicht zuletzt erscheint.
– Identität des jeweiligen Gesichts (entweder abgeglichen mit in vorherigen Szenen gesehenen Gesichtern, oder abgeglichen mit einer Gesichtsdatenbank) – dies erfordert auch eine gewisse Gesichtserkennung.

[0167] Der Nachführungsalgorithmus verwendet die Ergebnisse des Gesichtserkennungsalgorithmus, die unabhängig an jedem Vollbild der Bildfolge ablaufen, an seinem Startpunkt. Weil der Gesichtserkennungsalgorithmus manchmal Gesichter verfehlen (nicht erkennen) kann, ist ein gewisses Verfahren des Interpolierens der fehlenden Gesichter nützlich. Diesbezüglich wurde ein Kalman-Filter verwendet, um die nächste Position des Gesichts vorherzusagen, und ein Hautfarben-Abgleichalgorithmus wurde verwendet, um bei der Nachführung der Gesichter zu helfen. Außerdem ist, weil der Gesichtserkennungsalgorithmus häufig zu falschen Annahmen führt, ein gewisses Verfahren zu ihrem Zurückweisen ebenfalls nützlich.

[0168] Der Algorithmus ist schematisch in Fig. 22 dargestellt.

[0169] Der Algorithmus wird nachfolgend im Detail beschrieben, aber zusammenfassend werden eingegebene Videodaten **554** (die Bildfolge darstellend) einem Gesichtsdetektor der in dieser Anmeldung beschriebenen Art und einem Hautfarbenabgleichdetektor **550** zugeführt. Der Gesichtsdetektor versucht, eines oder mehrere Gesichter in jedem Bild zu erkennen. Wenn ein Gesicht erkannt wird, wird ein Kalman-Filter **560** eingerichtet, um der Position dieses Gesichts zu folgen. Das Kalman-Filter erzeugt eine Vorhersageposition für das gleiche Gesicht im nächsten Bild in der Folge. Ein Augenpositionsvergleicher **570**, **580** erfasst, ob der Gesichtsdetektor **540** ein Gesicht an dieser Position (oder innerhalb eines bestimmten Schwellenwertabstands dieser Position) im nächsten Bild erkennt. Falls festgestellt wird, dass dies der Fall ist, dann wird diese erkannte Gesichtsposition verwendet, um das Kalman-Filter zu aktualisieren, und der Prozess wird fortgesetzt.

[0170] Falls bei oder nahe der Vorhersageposition kein Gesicht erkannt wird, dann wird ein Hautfarbenabgleichverfahren **550** verwendet. Dies ist eine weniger genaue Gesichtserkennungstechnik, die so eingerichtet ist, dass sie einen geringeren Schwellenwert der Akzeptanz als der Gesichtsdetektor **540** hat, sodass es für die Hautfarbenabgleichtechnik möglich ist, ein Gesicht (was als ein Gesicht angesehen wird) selbst dann an dieser Position zu erfassen, wenn der Gesichtsdetektor keine positive Erkennung machen kann. Falls durch den Hautfarbenabgleich auf „Gesicht" erkannt wird, wird seine Position an das Kalman-Filter als eine aktualisierte Position gegeben, und der Prozess wird fortgesetzt.

[0171] Falls weder durch den Gesichtsdetektor **450** noch durch den Hauptfarbendetektor **550** eine Übereinstimmung festgestellt wird, dann wird die Vorhersageposition verwendet, um das Kalman-Filter zu aktualisieren.

[0172] Alle diese Ergebnisse werden Annahmekriterien unterzogen (siehe unten). So werden z.B. ein Gesicht, das auf der Basis einer positiven Erkennung durch eine Folge verfolgt wird, und der Rest als Vorhersagen oder der Rest als Hautfarbenerkennungen zurückgewiesen.

[0173] Ein separates Kalman-Filter wird benutzt, um jedes Gesicht in dem Nachführungsalgorithmus zu verfolgen.

[0174] Um ein Kalman-Filter zum Nachführen eines Gesichts zu benutzen, muss ein das Gesicht darstellendes Zustandsmodell erzeugt werden. In dem Modell wird die Position des jeweiligen Gesichts durch einen 4-dimensionalen Vektor mit den Koordinaten des linken und des rechten Auges dargestellt, welche ihrerseits durch eine vorbestimmte Beziehung zur Mittenposition des Fensters abgeleitet werden, wobei der verwendete Maßstab

$$p(k) = \begin{bmatrix} FirstEyeX \\ FirstEyeY \\ SecondEyeX \\ SecondEyeY \end{bmatrix}$$

ist, wobei k die Vollbildnummer ist.

[0175] Der aktuelle Zustand des Gesichts wird aus seiner Position, Geschwindigkeit und Beschleunigung in einem 12-dimensionalen Vektor dargestellt:

$$\hat{z}(k) = \begin{bmatrix} p(k) \\ \dot{p}(k) \\ \ddot{p}(k) \end{bmatrix}$$

Erstes Gesicht erkannt

[0176] Der Nachführungsalgorithmus tut nichts, bis er ein Vollbild mit einem Gesichtserkennungsergebnis empfängt, das angibt, dass ein Gesicht vorhanden ist.

[0177] Ein Kalman-Filter wird dann für jedes erkannte Gesicht in diesem Vollbild initialisiert. Sein Zustand wird mit der Position des Gesichts und mit null Geschwindigkeit und Beschleunigung initialisiert:

$$\hat{z}_a(k) = \begin{bmatrix} p(k) \\ 0 \\ 0 \end{bmatrix}$$

[0178] Es werden auch einige weitere Attribute zugewiesen: Die Zustandsmodellfehlerkovarianz Q und die Kovarianz R. Die Fehlerkovarianz des Kalman-Filters P wird ebenfalls initialisiert. Diese Parameter werden nachfolgend in mehr Einzelheiten beschrieben. Zu Beginn des folgenden Vollbildes und jedes nachfolgenden Vollbildes wird ein Kalman-Filtervorhersageprozess ausgeführt.

Kalman-Filtervorhersageprozess

[0179] Für jedes existierende Kalman-Filter wird die nächste Position des Gesichts mittels der nachfolgend dargestellten Standard-Kalman-Filtervorhersagegleichungen vorhergesagt. Das Filter verwendet den vorherigen Zustand (bei Vollbild k – 1) und einige weitere interne und externe Variablen, um den aktuellen Zustand des Filters (bei Vollbild k) abzuschätzen.

Zustandsvorhersagegleichung: $\hat{z}_b(k) = \Phi(k, k-1)\hat{z}_a(k-1)$

Kovarianzvorhersagegleichung: $P_b(k) = \Phi(k, k-1)P_a(k-1)\Phi(k, k-1)^T + Q(k)$

wobei $\hat{z}_b(k)$ den Zustand vor dem Aktualisieren des Filters für das Vollbild k bezeichnet, $\hat{z}_a(k-1)$ den Zustand nach dem Aktualisieren des Filters für das Vollbild k – 1 (oder den initialisierten Zustand, falls es ein neues Filter ist) bezeichnet, und $\Phi(k, k-1)$ die Zustandsübergangsmatrix ist. Es wurde mit verschiedenen Zustandsübergangsmatrixen experimentiert, wie nachfolgend beschrieben. In ähnlicher Weise bezeichnet $P_b(k)$ die Fehlerkovarianz des Filters vor dem Aktualisieren des Filters für ein Vollbild k, und $P_a(k-1)$ bezeichnet die Fehlerkovarianz des Filters nach dem Aktualisieren des Filters für das vorherige Vollbild (oder den initialisierten Wert, falls es ein neues Filter ist). $P_b(k)$ kann als eine interne Variable des Filters angesehen werden, die seine Genauigkeit modelliert.

[0180] Q(k) ist die Fehlerkovarianz des Zustandsmodells. Ein hoher Wert von Q(k) bedeutet, dass für die vorhergesagten Werte des Filterzustands (d.h. der Gesichtsposition) angenommen wird, dass sie einen hohen Fehlerpegel haben. Durch Einstellen dieses Parameters kann das Verhalten des Filters verändert und potenziell für die Gesichtserkennung verbessert werden.

Zustandsübergangsmatrix

[0181] Die Zustandsübergangsmatrix $\Phi(k,k-1)$ bestimmt, wie die Vorhersage des nächsten Zustands erfolgt. Mittels der Gleichungen für die Bewegung kann die folgende Matrix für $\Phi(k,k-1)$ abgeleitet werden:

$$\Phi(k,k-1)=\begin{bmatrix} I_4 & I_4\Delta t & \frac{1}{2}I_4(\Delta t)^2 \\ O_4 & I_4 & I_4\Delta t \\ O_4 & O_4 & I_4 \end{bmatrix}$$

wobei O_4 eine 4×4-Nullmatrix ist und I_4 eine 4×4-Identitätsmatrix ist. Δt kann einfach auf 1 gesetzt werden (d.h. Einheiten von t sind Vollbildperioden).

[0182] Diese Zustandsübergangsmatrix modelliert Position, Geschwindigkeit und Beschleunigung. Es wurde jedoch herausgefunden, dass die Verwendung der Beschleunigung dazu tendierte, die Gesichtsvorhersagen zur Kante des Bildes beschleunigen zu lassen, wenn keine Gesichtserkennungen verfügbar waren, um den vorhergesagten Zustand zu korrigieren. Deshalb wurde eine einfachere Zustandsübergangsmatrix ohne Verwendung der Beschleunigung bevorzugt:

$$\Phi(k,k-1)=\begin{bmatrix} I_4 & I_4\Delta t & O_4 \\ O_4 & I_4 & O_4 \\ O_4 & O_4 & I_4 \end{bmatrix}$$

[0183] Die vorhergesagten Augenpositionen jedes Kalman-Filters $\hat{z}_b(k)$ werden mit allen Gesichtserkennungsergebnissen im aktuellen Vollbild (falls es welche gibt) verglichen. Falls der Abstand zwischen den Augenpositionen unter einem gegebenen Schwellenwert liegt, dann kann angenommen werden, dass die Gesichtserkennung zum gleichen Gesicht gehört, welches durch das Kalman-Filter modelliert wird. Das Gesichtserkennungsergebnis wird dann als eine Beobachtung y(k) des aktuellen Zustands des Gesichts behandelt:

$$y(k)=\begin{bmatrix} p(k) \\ 0 \\ 0 \end{bmatrix}$$

wobei p(k) die Position der Augen in dem Gesichtserkennungsergebnis ist. Diese Beobachtung wird während der Aktualisierungsstufe des Kalman-Filters benutzt, um die Vorhersage korrigieren zu helfen.

Hautfarbenabgleich

[0184] Ein Hautfarbenabgleich wird nicht für Gesichter verwendet, die erfolgreich zu den Gesichtserkennungsergebnissen passen. Ein Hautfarbenabgleich wird nur für Gesichter durchgeführt, deren Position durch das Kalman-Filter vorhergesagt worden ist, aber die kein passendes Gesichtserkennungsergebnis im aktuellen Vollbild und deshalb keine Beobachtungsdaten, um das Kalman-Filter aktualisieren zu helfen, haben.

[0185] In einer ersten Technik wird für jedes Gesicht ein elliptischer Bereich, zentriert um die vorherige Person des Gesichts, aus dem vorherigen Vollbild extrahiert. Ein Beispiel eines solchen Bereichs **600** in dem Gesichtsfenster **610** ist schematisch in Fig. 24 dargestellt. Ein Farbenmodell wird mittels der Farbwertedaten aus diesem Bereich angelegt, um eine Abschätzung des Mittelwerts und der Kovarianz der Cr- und der Cb-Werte basierend auf einem Gauß-Modell zu erzeugen.

[0186] Ein Bereich um die vorhergesagte Gesichtsposition im aktuellen Vollbild wird dann durchsucht, und die Position, die am besten zu dem Farbmodell passt, wieder gemittelt über einen elliptischen Bereich, wird ausgewählt. Falls der Farbentreffer ein gegebenes Ähnlichkeitskriterium erfüllt, dann wird diese Position als eine Beobachtung y(k) des aktuellen Zustands des Gesichts in der gleichen Weise wie für die Gesichtserkennungsergebnisse im vorherigen Abschnitt beschrieben, verwendet.

[0187] Fig. 23a und Fig. 23b zeigen schematisch die Erzeugung des Suchbereichs. Insbesondere zeigt Fig. 23a schematisch die Vorhersageposition **620** des Gesichts im nächsten Bild **630**. Beim Hautfarbenabgleich wird ein Suchbereich **640** um die Vorhersageposition **620** im nächsten Bild für das Gesicht durchsucht.

[0188] Falls der Farbenabgleich nicht das Ähnlichkeitskriterium erfüllt, dann sind keine zuverlässigen Beobachtungsdaten für das aktuelle Vollbild verfügbar. Stattdessen wird der vorhergesagte Zustand $\hat{z}_b(k)$ als Beobachtung verwendet:

$y(k) = \hat{z}_b(k)$

[0189] Die oben beschriebenen Hautfarbenabgleichverfahren benutzen ein einfaches Gauß-Hautfarbenmodell. Das Modell wird auf einem elliptischen Bereich, zentriert auf dem Gesicht im vorherigen Vollbild, angelegt und benutzt, um den am besten passenden elliptischen Bereich im aktuellen Vollbild zu finden. Um jedoch eine potenziell bessere Leistung vorzusehen, wurden zwei weitere Verfahren entwickelt: ein Farbhistogrammverfahren und ein Farbmaskenverfahren. Diese werden nun beschrieben.

Farbhistogrammverfahren

[0190] In diesem Verfahren wird anstelle der Verwendung eines Gauß-Modells für die Farbverteilung im verfolgten Gesicht ein Farbhistogramm verwendet.

[0191] Für jedes verfolgte Gesicht im vorherigen Vollbild wird ein Histogramm aus Cr- und Cb-Werten in einem quadratischen Fenster um das Gesicht berechnet. Hierzu werden für jedes Pixel die Cr- und Cb-Werte zuerst zu einem einzelnen Wert kombiniert. Ein Histogramm wird dann berechnet, das die Frequenz des Auftretens dieser Werte im gesamten Fenster misst. Weil die Anzahl kombinierter Cr- und Cb-Werte groß ist (256×256 mögliche Kombinationen), werden die Werte quantisiert, bevor das Histogramm berechnet wird.

[0192] Nach dem Berechnen eines Histogramms für ein verfolgtes Gesicht im vorherigen Vollbild wird das Histogramm im aktuellen Vollbild benutzt, um zu versuchen, die wahrscheinlichste neue Position des Gesichts abzuschätzen, indem der Bereich des Bildes mit der ähnlichsten Farbverteilung gefunden wird. Wie schematisch in **Fig. 23a** und **Fig. 23b** dargestellt, erfolgt dies durch Berechnen eines Histogramms in exakt der gleichen Weise wie für den Bereich von Fensterpositionen in einem Suchbereich des aktuellen Vollbildes. Dieser Suchbereich überdeckt einen gegebenen Bereich um die vorhergesagte Gesichtsposition. Die Histogramme werden dann durch Berechnen des mittleren quadratischen Fehlers (MSE) zwischen dem ursprünglichen Histogramm für das verfolgte Gesicht im vorherigen Vollbild und jedem Histogramm im aktuellen Vollbild verglichen. Die abgeschätzte Position des Gesichts im aktuellen Vollbild wird durch die Position des minimalen MSE gegeben.

[0193] Verschiedene Modifikationen können an diesem Algorithmus vorgenommen werden, einschließlich:
- Verwenden von drei Kanälen (Y, Cr und Cb) anstelle von zwei (Cr, Cb).
- Variieren der Anzahl der Quantisierungsniveaus.
- Teilen des Fensters in Blöcke und Berechnen eines Histogramms für jeden Block. Auf diese Weise wird das Farbhistogrammverfahren positionsabhängig. Der MSE zwischen jedem Paar von Histogrammen wird in diesem Verfahren aufsummiert.
- Variieren der Anzahl der Blöcke, in welche das Fenster aufgeteilt ist.
- Variieren der Blöcke, die aktuell verwendet werden – z.B. Verzichten auf die äußeren Blöcke, die nur teilweise Gesichtspixel enthalten können.

[0194] Für die in empirischen Versuchen dieser Technik verwendeten Testdaten wurden die besten Ergebnisse mittels der folgenden Bedingungen erzielt, obwohl andere Sätze von Bedingungen gleichermaßen gute oder bessere Ergebnisse mit anderen Testdaten vorhersehen können:
- 3 Kanäle (Y, Cr und Cb).
- 8 Quantisierungsniveaus für jeden Kanal (d.h. ein Histogramm enthält 8 × 8 × 8 = 512 Fächer).
- Aufteilen des Fensters in 16 Blöcke.
- Verwenden aller 16 Blöcke.

Farbmaskenverfahren

[0195] Dieses Verfahren basiert auf dem oben zuerst beschriebenen Verfahren. Es verwendet ein Gauß-Hautfarbenmodell, um die Verteilung der Pixel im Gesicht zu beschreiben.

[0196] In dem oben zuerst beschriebenen Verfahren wird ein elliptischer Bereich, zentriert auf dem Gesicht, verwendet, um die Gesichter farbabzugleichen, da angenommen wird, dass dies die Menge der Hintergrundpixel reduziert oder minimiert, welche das Modell verschlechtern können.

[0197] Im vorliegenden Farbmaskenmodell wird nach wie vor ein ähnlicher elliptischer Bereich benutzt, um ein Farbenmodell an dem ursprünglich verfolgten Gesicht im vorherigen Vollbild anzulegen, z.B. durch Anwenden des Mittelswerts und der Kovarianz von RGB oder YCrCb auf Einstellparameter eines Gauß-Modells (oder

alternativ kann ein Vorgabefarbenmodell, wie beispielsweise ein Gauß-Modell verwendet werden, siehe unten). Es wird jedoch nicht beim Suchen nach dem besten Abgleich im aktuellen Vollbild verwendet. Stattdessen wird ein Maskenbereich basierend auf der Verteilung der Pixel im ursprünglichen Gesichtsfenster aus dem vorherigen Vollbild berechnet. Die Maske wird durch Finden der 50% der Pixel im Fenster, die am besten zum Farbenmodell passen, berechnet. Ein Beispiel ist in Fig. 25a bis Fig. 25c dargestellt. Insbesondere zeigt Fig. 25a schematisch das Ausgangsfenster im Test; Fig. 25b zeigt schematisch das elliptische Fenster, das zum Anlegen des Farbenmodells verwendet wird; und Fig. 25c zeigt schematisch die durch die 50% der Pixel, die am besten zum Farbenmodell passen, definierte Maske.

[0198] Um die Position des Gesichts im aktuellen Vollbild abzuschätzen, wird ein Suchbereich um die vorhergesagte Gesichtsposition durchsucht (wie vorher), und der „Abstand" vom Farbenmodell wird für jedes Pixel berechnet. Der „Abstand" bezieht sich auf eine Differenz zum Mittelwert, normiert in jeder Dimension durch die Varianz in dieser Dimension. Ein Beispiel des resultierenden Abstandsbildes ist in Fig. 26 gezeigt. Für jede Position in dieser Abstandsabbildung (oder für einen reduzierten Satz von abgetasteten Positionen, um die Berechnungszeit zu verringern), werden die Pixel des Abstandsbildes über einen maskenförmigen Bereich gemittelt. Die Position mit dem niedrigsten mittleren Abstand wird dann als beste Abschätzung für die Position des Gesichts in diesem Vollbild ausgewählt.

[0199] Dieses Verfahren unterscheidet sich von dem ursprünglichen Verfahren somit darin, dass ein maskenförmiger Bereich in dem Abstandsbild verwendet wird, anstelle eines elliptischen Bereichs. Dies lässt das Farbabgleichverfahren sowohl Farb-, als auch Forminformationen benutzen.

[0200] Zwei Varianten werden für die Techniken vorgeschlagen und wurden in empirischen Versuchen realisiert:

(a) Ein Gauß-Hautfarbenmodell wird mittels des Mittelwerts und der Kovarianz von Cr und Cb aus einem elliptischen Bereich, zentriert auf dem verfolgten Gesicht, im vorherigen Vollbild angelegt.

(b) Ein Vorgabe-Gauß-Hautfarbenrriodell wird verwendet, um sowohl die Maske im vorherigen Rahmen zu berechnen als auch das Abstandsbild im aktuellen Rahmen zu berechnen.

[0201] Es wird nun die Verwendung der Gauß-Hautfarbenmodelle weiter beschrieben. Ein Gauß-Modell für die Hautfarbenklasse wird mittels der Farbwertekomponenten des YCbCr-Farbraums gebildet. Die Ähnlichkeit der Testpixel zur Hautfarbenklasse kann dann gemessen werden. Dieses Verfahren sieht eine Hautfarbenwahrscheinlichkeitsabschätzung für jeden Pixel vor, unabhängig von den eigengesichtsbasierten Ansätzen.

[0202] Es sei w der Vektor der CbCr-Werte eines Testpixels. Die Wahrscheinlichkeit, dass w zur Hautfarbenklasse S gehört, wird durch einen zweidimensionalen Gauß nachgebildet:

$$p(w|S) = \frac{\exp\left\{-\frac{1}{2}(w-\mu_S)\sum_S^{-1}(w-\mu_S)\right\}}{2\pi|\Sigma_S|^{1/2}}$$

wobei der Mittelwert μ_S und die Kovarianzmatrix Σ_S der (vorherigen) Verteilung aus einem Trainingssatz von Hautfarbenwerten abgeschätzt werden.

[0203] Die Hautfarbenerkennung wird nicht als ein effektiver Gesichtsdetektor angesehen, wenn sie allein verwendet wird. Dies deshalb, weil es viele Bereiche eines Bildes geben kann, die ähnlich einer Hautfarbe, aber nicht notwendigerweise Gesichter, sondern z.B. andere Körperteile sind. Sie kann jedoch benutzt werden, um das Leistungsvermögen der eigenblock-basierten Ansätze durch Verwenden eines kombinierten Ansatzes, wie bezüglich des vorliegenden Gesichtsnachführungssytems beschrieben, zu verbessern. Die Entscheidungen darüber, ob die im Gesicht erkannten Augenpositionen oder die farbabgeglichenen Augenpositionen wie bei der Beobachtung für das Kalman-Filter angenommen werden oder ob keine Beobachtung akzeptiert wird, werden gespeichert. Diese werden später benutzt, um die weitere Gültigkeit der durch jedes Kalman-Filter modellierten Gesichter zu bemessen.

[0204] Varianten des Farbmaskenverfahrens werden nun unter Bezug auf Fig. 27 und Fig. 28 beschrieben.

Kalman-Filter-Aktualisierungsschritt

[0205] Der Aktualisierungsschritt wird benutzt, um eine geeignete Ausgabe des Filters für das aktuelle Vollbild

basierend auf der Zustandsvorhersage und den Beobachtungsdaten zu bestimmen. Er aktualisiert auch die internen Variablen des Filters basierend auf dem Fehler zwischen dem vorhergesagten Zustand und dem beobachteten Zustand.

[0206] Die folgenden Gleichungen werden im Aktualisierungsschritt verwendet:

Kalman-Verstärkungsgleichung $K(k) = P_b(k)H^T(k)[H(k)P_b(k)H^T(k) + R(k)]^{-1}$

Zustandsaktualisierungsgleichung $\hat{z}_a(k) = \hat{z}_b(k) + K(k)[y(k) - H(k)\hat{z}_b(k)]$

Kovarianz-Aktualisierungsgleichung $P_a(k) = P_b(k) - K(k)H(k)P_b(k)$

[0207] Hierbei bezeichnet K(k) den Kalman-Verstärkungsfaktor, eine weitere interne Variable des Kalman-Filters. Er wird benutzt, um zu bestimmen, um wie viel der vorhergesagte Zustand basierend auf dem beobachteten Zustand y(k) eingestellt werden sollte.

[0208] H(k) ist die Beobachtungsmatrix. Sie bestimmt, welche Teile des Zustands beobachtet werden können. In unserem Fall kann nur die Position des Gesichts beobachtet werden, nicht seine Geschwindigkeit oder Beschleunigung, sodass die folgende Matrix für H(k) verwendet wird:

$$H(k) = \begin{bmatrix} I_4 & O_4 & O_4 \\ O_4 & O_4 & O_4 \\ O_4 & O_4 & O_4 \end{bmatrix}$$

[0209] R(k) ist die Fehlerkovarianz der Beobachtungsdaten. In einer ähnliche Weise wie bei Q(k) bedeutet ein hoher Wert von R(k), dass die beobachteten Werte des Filterzustands (d.h. die Gesichtserkennungsergebnisse oder Farbabgleiche) als einen hohen Fehlerpegel aufweisend angenommen werden. Durch Einstellen dieses Parameters kann das Verhalten des Filters verändert werden und potenziell für die Gesichtserkennung verbessert werden. Für unsere Erkenntnisse wurde ein relativ zu Q(k) großer von R(k) als geeignet herausgefunden (dies bedeutet, dass die vorhergesagten Gesichtspositionen als zuverlässiger als die Beobachtungen angesehen werden). Man beachte, dass es erlaubt ist, diese Parameter von Vollbild zu Vollbild zu variieren. Deshalb kann es ein interessierender Bereich zukünftiger Untersuchungen sein, die Relativwerte von R(k) und Q(k) in Abhängigkeit davon einzustellen, ob die Beobachtung auf einem Gesichtserkennungsergebnis (zuverlässig) oder einem Farbabgleich (weniger zuverlässig) basiert.

[0210] Für jedes Kalman-Filter wird der aktualisierte Zustand $\hat{z}_a(k)$ als Endentscheidung über die Position des Gesichts verwendet. Diese Daten werden an eine Datei ausgegeben und gespeichert.

[0211] Unpassende Gesichtserkennungsergebnisse werden als neue Gesichter behandelt. Ein neues Kalman-Filter wird für jedes von ihnen initialisiert. Gesichter werden entfernt, welche

– den Rand des Bildes verlassen, und/oder

– eine fehlende weitere, sie unterstützende Evidenz haben (wenn es einen hohen Anteil von Beobachtungen basierend auf Kalman-Filtervorhersagen gegenüber Gesichtserkennungsergebnissen oder Farbabgleichen gibt).

[0212] Für diese Gesichter wird das zugehörige Kalman-Filter entfernt, und es werden keine Daten an eine Datei ausgegeben. Als ein optionaler Unterschied zu diesem Ansatz können, wenn ein Gesicht erfasst wird, welches das Bild verlässt, die Nachführungsergebnisse bis zu dem Rahmen vor dem Verlassen des Bildes gespeichert und als gültige Gesichtserkennungsergebnisse behandelt werden (vorausgesetzt, dass die Ergebnisse weitere Kriterien zum Validieren von Nachführungsergebnissen erfüllen).

[0213] Diese Regeln können durch Einbringen einiger zusätzlichen Variablen formalisiert und aufgebaut werden:

prediction_acceptance_ratio_threshold Falls während der Verfolgung eines gegebenen Gesichts der Anteil akzeptierter Kalman-vorhergesagter Gesichtspositionen diesen Schwellenwert übersteigt, dann wird das verfolgte Gesicht zurückgewiesen. Dieser ist derzeit auf 0,8 gesetzt.

detection_acceptance_ratio_threshold Während eines letzten Durchgangs durch alle Vollbilder wird, falls für ein gegebenes Gesicht der Anteil akzeptierter Gesichtserkennungen unter diesen Schwellenwert fällt, dann das verfolgte Gesicht zurückgewiesen. Dieser ist derzeit auf 0,08 gesetzt.

min_frames Während eines letzten Durchgangs durch alle Vollbilder wird, falls für ein gegebenes Gesicht die Anzahl Auftritte geringer als min_frames ist, das Gesicht zurückgewiesen. Dies tritt wahrscheinlich nur nahe dem Ende einer Folge auf. min_frames ist derzeit auf 5 gesetzt.
final_prediction_acceptance_ratio_threshold und min_frames2 Während eines letzten Durchgangs durch alle Vollbilder wird, falls für ein gegebenes verfolgtes Gesicht die Anzahl Auftritte geringer als min_frames2 ist, UND der Anteil akzeptierter Kalman-vorhergesagter Gesichtspositionen final_prediction_acceptance_ratio_threshold übersteigt, das Gesicht zurückgewiesen. Wieder tritt dies wahrscheinlich nur nahe dem Ende einer Folge auf. final_rediction_acceptunce_ratio_threshold ist derzeit auf 0,5 gesetzt, und min-frames2 ist derzeit auf 10 gesetzt.
min_eye_spacing Außerdem werden Gesichter nun entfernt, falls sie so verfolgt werden, dass der Augenabstand unter einen gegebenen minimalen Abstand sinkt. Dies kann passieren, falls das Kalman-Filter fälschlicherweise glaubt, dass der Augenabstand kleiner wird und es keine andere Evidenz, z.B. Gesichtserkennungsergebnisse, gibt, um diese Annahme zu korrigieren. Falls nicht korrigiert, würde der Augenabstand schließlich zu Null werden. Als eine optionale Alternative kann ein Minimum oder eine untere Grenze der Augentrennung gesetzt werden, sodass, falls die erfasste Augentrennung auf die minimale Augentrennung sinkt, der Erkennungsprozess fortgesetzt wird, um nach Gesichtern mit dieser Augentrennung zu suchen, aber nicht einer kleineren Augentrennung.

[0214] Es ist zu beachten, dass der Nachführungsprozess nicht auf eine Nachführung durch eine Videofolge in einer zeitlichen Vorwärtsrichtung beschränkt ist. Angenommen dass die Bilddaten zugreifbar bleiben (d.h. der Prozess ist nicht in Echtzeit oder die Bilddaten werden für eine vorübergehend kontinuierliche Nutzung gepuffert), könnte der gesamte Nachführungsprozess in einer zeitlichen Rückwärtsrichtung ausgeführt werden. Oder wenn eine erste Gesichtserkennung erfolgt (häufig teilweise durch eine Videosequenz, könnte der Nachführungsprozess in beiden zeitlichen Richtungen initiiert werden. Als eine weitere Option könnte der Nachführungsprozess in beiden zeitlichen Richtungen durch eine Videosequenz laufen, wobei die Ergebnisse so kombiniert werden, dass (z.B.) ein die Akzeptanzkriterien erfüllendes verfolgtes Gesicht als ein gültiges Ergebnis enthalten ist, in welcher Richtung auch immer die Nachführung stattfindet.

[0215] Einige weitere Regeln, die bezüglich der Gesichtsnachführung angewendet werden können, werden nun beschrieben.

Alternative/zusätzliche Regeln zum Starten/Halten einer Gesichtsnachführung

[0216] In den oben beschriebenen Verfahren wird eine Gesichtsnachführung für jede Gesichtserkennung gestartet, die nicht mit einer existierenden Nachführung abgeglichen werden kann. Dies kann zu vielen falschen Erkennungen führen, die irrtümlicherweise verfolgt werden und für mehrere Vollbilder bestehen bleiben, bevor sie schließlich durch eine der existierenden Regeln (z.B. prediction_acceptance_ratio_threshold) zurückgewiesen werden.

[0217] Ebenso sind die Regeln zum Zurückweisen einer Verfolgung (z.B. prediction_acceptance_ratio_threshold, detection_acceptance_ratio_threshold) gegen eine Nachführung einer Person, die ihren Kopf zur Seite dreht, eine bestimmte Zeitlänge vorgespannt. In der Realität ist es häufig nützlich, jemanden weiterzuverfolgen, der dies tut.

[0218] Der erste Teil der Lösung hilft, zu verhindern, dass falsche Erkennungen fehlerhafte Spuren absetzen.

[0219] Eine Gesichtsnachführung wird nach wie vor intern für jede Gesichtserkennung gestartet, die zu keiner existierenden Verfolgung passt. Sie wird jedoch nicht vom Algorithmus ausgegeben.

[0220] Damit diese Spur gehalten wird, müssen die ersten f Vollbilder in der Spur Gesichtserkennungen, d.h. vom Typ D sein. Falls alle ersten f Rahmen vom Typ D sind, dann wird die Spur gehalten und Gesichtspositionen werden von dem Algorithmus von dem Vollbild f an ausgegeben. Falls alle ersten n Rahmen nicht vom Typ D sind, dann wird die Gesichtsnachführung beendet und keine Gesichtspositionen werden für diese Nachführung ausgegeben. f ist typischerweise auf 2, 3 oder 5 gesetzt.

[0221] Der zweite Teil der Lösung erlaubt eine Verfolgung von Gesichtern im Profil für eine lange Dauer, anstatt dass ihre Spuren wegen eines niedrigen detection_acceptance_ratio beendet werden. prediction_acceptance_ratio_threshold und detection_acceptance_ratio_threshold werden in diesem Fall nicht eingeschaltet. Stattdessen ist es eine Option, das folgende Kriterium einzuschließen, um eine Gesichtsspur zu halten:

g fortlaufende Gesichtserkennungen sind alle n Vollbilder erforderlich, um die Gesichtsspur zu halten, wobei g typischerweise auf einen ähnlichen Wert wie f, z.B. 1 bis 5 Vollbilder, gesetzt ist und n der maximalen Anzahl an Vollbildern entspricht, für welche wir jemanden verfolgen können wollen, wenn sie von der Kamera für zum Beispiel 10 Sekunden (= 250 oder 300 Vollbilder in Abhängigkeit von der Vollbildrate) weggedreht werden.

[0222] Dies kann auch mit den prediction_acceptance_ratio_threshold und detection_acceptance_ratio_threshold – Regeln kombiniert werden. Alternativ können prediction_acceptance_ratio_threshold und detection_acceptance_ratio_threshold auf einer laufenden Basis z.B. über nur die letzten 30 Vollbilder statt seit dem Beginn der Nachführung angewendet werden.

Schlechte-Farben-Schwellenwert-Regel

[0223] In den oben beschriebenen Verfahren wird die Hautfarbe des Gesichts nur während der Hautfarbennachführung geprüft. Dies bedeutet, dass nicht-hautfarbige falsche Erkennungen verfolgt werden können oder die Gesichtsverfolgung mittels der vorhergesagten Gesichtsposition in nicht-hautfarbige Positionen abwandern kann.

[0224] Um sich darum zu kümmern, wird, für welche Akzeptanzart des Gesichts auch immer (Erkennung, Hautfarbe oder Kalman-Vorhersage), seine Hautfarbe geprüft. Falls sein Abstand von der Hautfarbe einen bad_colour_threshold übersteigt, dann wird die Gesichtsverfolgung beendet.

[0225] Ein effizienter Weg, dies zu realisieren, ist es, den Abstand von der Hautfarbe jedes während der Hautfarbennachführung berechneten Pixels zu benutzen. Falls diese über den Gesichtsbereich gemittelte Messung (entweder über einen maskenförmigen Bereich, über einen elliptischen Bereich oder über das gesamte Gesichtsfenster, in Abhängigkeit davon, welches Hautfarbennachführungsverfahren verwendet wird) einen festen Schwellenwert überschreitet, dann wird die Gesichtsverfolgung beendet.

Zulässigkeitsprüfung mit einem Bereich einer interessierenden Abbildung

[0226] In anderen hier beschriebenen Verfahren wird ein Bereich eines interessierenden Preprozessors benutzt, um sicherzustellen, welche Bereiche des Bildes eine nicht-gesichtsartige Varianz besitzen. Dies wird für jeden Maßstab wiederholt, und diese Bereiche werden dann aus der Gesichtserkennungsverarbeitung weggelassen. Es ist jedoch nach wie vor für ein hautfarben-verfolgtes oder ein Kalman-vorhergesagtes Gesicht möglich, sich in einen (nicht-gesichtsartigen) niedrigen oder hohen Varianzbereich des Bildes zu bewegen.

[0227] Um sich darum zu kümmern, werden während der Varianz-Vorverarbeitung die Varianzwerte (oder Gradientenwerte) für die Bereiche um existierende Gesichtsverfolgungen gespeichert.

[0228] Wenn die Endentscheidung über die nächste Position des Gesichts gemacht ist (mit irgendeinem Akzeptanztyp, entweder Gesichtserkennung, Hautfarbe oder Kalman-Vorhersage), wird die Position gegen die gespeicherten Varianz- (oder Gradienten-) Werte in dem Bereich der interessierenden Abbildung auf Zulässigkeit geprüft. Falls festgestellt wird, dass die Stelle eine sehr hohe oder sehr niedrige Varianz (oder Gradienten) hat, wird sie als nicht-gesichtsartig angesehen und die Gesichtsverfolgung wird beendet. Dies vermeidet Gesichtsverfolgungen, die in Hintergrundbereiche niedriger (oder hoher) Varianz des Bildes abwandern. Alternativ wird die Varianz der neuen Gesichtsstelle frisch berechnet (nützlich, falls eine Varianzvorverarbeitung nicht verwendet wird).

[0229] Man beachte, dass die verwendete Varianzmessung entweder eine herkömmliche Varianz oder die Summe von Unterschieden benachbarter Pixel (Gradient) oder irgendeine andere Varianztypmessung sein kann.

[0230] In dem schematisch in Fig. 22 dargestellten Nachführungssystem sind drei weitere Merkmale enthalten.

[0231] Aufnahmegrenzdaten **560** (aus zu der geprüften Bildfolge gehörenden Metadaten; oder in der Kamera von Fig. 2 erzeugte Metadaten) definieren die Grenzen jeder sequenziellen „Aufnahme" in der Bildfolge. Das Kalman-Filter wird an den Aufnahmegrenzen zurückgesetzt und darf keine Vorhersage zu einer nachfolgenden Aufnahme hinübertragen, da die Vorhersage bedeutungslos wäre.

[0232] Nutzer-Metadaten **542** und Kameraeinstellungs-Metadaten **544** werden als Eingaben dem Gesichtsdetektor **540** zugeführt. Diese können auch in einem Nicht-Nachführungssystem verwendet werden. Beispiele der Kameraeinstellungs-Metadaten wurden oben beschrieben. Nutzer-Metadaten können Informationen enthalten, wie beispielsweise:

– Art des Programms (z.B. Nachrichten, Interview, Schauspiel)

– Stückinformationen, wie beispielsweise die Bestimmung einer „fernen Aufnahme", einer „mittleren Großaufnahme", usw. (besondere Arten einer Kameraaufnahme führen zu einem unerwarteten Unterbereich von Gesichtsgrößen), wie viele Leute in jeder Aufnahme involviert sind (führt wieder zu einem unerwarteten Unterbereich von Gesichtsgrößen) und dergleichen.

– Sportbezogenen Informationen – Sport wird häufig von festen Kamerapositionen mittels Standardperspektiven und -aufnahmen gefilmt. Durch die Bestimmung davon in den Metadaten kann wieder ein Unterbereich von Gesichtsgrößen abgeleitet werden.

[0233] Die Art des Programms ist für die Art des Gesichts relevant, das in den Bildern oder der Bildfolge erwartet werden kann. Zum Beispiel würde man in einem Nachrichtenprogramm erwarten, ein einzelnes Gesicht für einen Hauptteil der Bildfolge zu sehen, das einen Bereich von (z.B.) 10% des Bildschirms bedeckt. Die Erkennung der Gesichter in unterschiedlichen Maßstäben kann als Reaktion auf diese Daten gewichtet werden, sodass Gesichtern von etwa dieser Größe eine erhöhte Wahrscheinlichkeit gegeben werden kann. Eine weitere Alternative oder ein zusätzlicher Ansatz besteht darin, dass der Suchbereich verkleinert wird, sodass statt der Suche nach Gesichtern in allen möglichen Maßstäben nur ein Untersatz von Maßstäben durchsucht wird. Dies kann die Verarbeitungsanforderungen des Gesichtserkennungsprozesses reduzieren. In einem software-basierten System, kann die Software schneller und/oder auf einem leistungsschwächeren Prozessor laufen. In einem hardware-basierten System (mit zum Beispiel einer anwendungsspezifischen integrierten Schaltung (ASIC) oder einem feldprogrammierbaren Gate Array (FPGA) – System) können die Hardware-Anforderungen reduziert werden.

[0234] Die anderen Arten von oben genannten Nutzer-Metadaten können ebenfalls in dieser Weise angewendet werden. Die Unterbereiche der „erwarteten Gesichtsgröße" können zum Beispiel in einer im Speicher **30** gehaltenen Nachschlagtabelle gespeichert werden.

[0235] Was die Kamera-Metadaten angeht, zum Beispiel die aktuellen Fokus- und Zoom-Einstellungen der Linse **110**, können diese ebenfalls den Gesichtsdetektor unterstützen, indem sie eine Anfangsangabe der erwarteten Bildgröße von irgendwelchen Gesichtern angeben, die im Vordergrund des Bildes vorhanden sein können. Diesbezüglich wird beachtet, dass die Fokus- und Zoom-Einstellungen zwischen ihnen die erwartete Trennung zwischen dem Camcorder **100** und einer gefilmten Person definieren, und auch die Vergrößerung des Objektivs **110**. Aus diesen zwei Attributen ist es basierend auf einer mittleren Gesichtsgröße möglich, die erwartete Größe (in Pixeln) eines Gesichts in den resultierenden Bilddaten zu berechnen, was wieder zu einem Unterbereich von Größen zur Suche oder einer Gewichtung der erwarteten Gesichtsgröße führt.

Vorteile des Nachführungsalgorithmus

[0236] Die Gesichtsnachführungstechnik hat drei Hauptvorteile:

– Sie erlaubt das Auffüllen verfehlter Gesichter mittels Kalman-Filterung und Hautfarben-Nachführung in Vollbilder, für die keine Gesichtserkennungsergebnisse verfügbar sind. Dies erhöht die wahre Akzeptanzrate über die Bildfolge.

– Sie stellt eine Gesichtsverknüpfung bereit: Durch erfolgreiches Verfolgen eines Gesichts weiß der Algorithmus automatisch, ob ein erkanntes Gesicht in einem zukünftigen Vollbild zur gleichen Person oder zu einer andern Person gehört. Somit können Szenen-Metadaten einfach aus diesem Algorithmus erzeugt werden, mit der Anzahl von Gesichtern in der Szene, den Vollbildern, für die sie vorhanden sind, und ein repräsentatives Verbrecherfoto jedes Gesichts bereitstellend.

– Falsche Gesichtserkennungen werden leicht zurückgewiesen, da solche Erkennungen dazu neigen, zwischen den Bildern nicht fortgesetzt zu werden.

[0237] Die oben beschriebenen Ausführungsbeispiele bezogen sich auf ein Gesichtserkennungssystem (einschließlich Trainings- und Erkennungsphasen) und mögliche Verwendungen dafür in einem Kamerarecorder und einem Editiersystem. Es ist selbstverständlich, dass es viele weitere mögliche Verwendungen solcher Techniken gibt, zum Beispiel (und nicht darauf beschränkt) Sicherheitsüberwachungssysteme, Medienbearbeitung allgemein (beispielsweise Videorecorder-Steuerungen), Videokonferenzsysteme und dergleichen.

Varianten der Farbmaskentechniken

[0238] Diese Varianten können einzeln oder, wo es technisch passend ist, in Kombination angewendet werden.

[0239] Variante 1: Ein Vorgabe-Gauß-Hautfarbenmodel wird verwendet, um den Farbabstandswert für jeden Pixel in dem Bild zu berechnen (um eine Farbabstandsabbildung zu erzielen). Wenn ein Gesicht erkannt wird, wird sein mittlerer Abstand von der Vorgabe über einen maskenförmigen Bereich berechnet. Bei der Zulässigkeitsprüfung zukünftiger Farben verfolgter Gesichter (oder erkannter Gesichter oder Kalman-vorhergesagter Gesichter) wird die Gesichtsverfolgung beendet, falls der Abstand von dem Vorgabe-Farbmodel sich außerhalb einer gegebenen Toleranz verändert. Dies bedeutet, dass (a) die gleichen Farbabstandswerte für alle Gesichtsverfolgungen verwendet werden können (es gibt keine Notwendigkeit für ein anderes Modell für jedes Gesicht, da wir ein Vorgabe-Farbenmodell anstelle des Anlegens einer Gesichtsfarbe verwenden) und (b) falls die Nachführung von einem Gesicht auf einen andersfarbigen Hintergrund abwandert, wird sie beendet, anstatt fortzudauern.

[0240] Variante 2: Anstelle eines Vorgabe-Farbmodells wird ein anderes Farbmodell für jedes Gesicht benutzt, angelegt auf diese Gesichtsfarbe, wenn es zuerst erkannt wird. Dann werden, wenn die Farbabstandabbildung berechnet wird, unterschiedliche Farbmodellparameter in unterschiedlichen Bereichen des Bildes in Anhängigkeit von der Position jedes Gesichts im vorherigen Vollbild verwendet. Die Farbabstandswerte können als eine gewichtete Summe auf Bereichen zwischen zwei Gesichtern miteinander vermischt werden. Dies erlaubt eine genauere Nachbildung der Farbennachführung bei jeder aktuellen Gesichtsfarbe, ohne mehr als einen Farbabstandswert für jede Pixelposition in dem Bild berechnen zu müssen.

[0241] Variante 3: Anstelle eines Vorgabe-Farbmodells oder unterschiedlicher Farbmodelle für jedes Gesicht wird ein allgemeines Farbmodell auf der mittleren Farbe aller erkannten Gesichter aus dem vorherigen Vollbild angelegt.

[0242] Variante 4: Beim Durchsuchen der Farbabstandsabbildung mit einem quadratischen kopfgroßen Fenster, um den minimalen mittleren Abstand aus der Hautfarbe zu finden, ergeben Bereiche innerhalb der Maske einen positiven Beitrag und Bereiche außerhalb der Maske einen negativen Beitrag. Dies bedeutet, dass einfarbige hautfarbene Bereiche einen Nettoabstand von der Hautfarbe von Null haben sollten. Es beutet auch, dass die Formenabgleicheigenschaften des Maskenverfahrens verstärkt werden. In dem oben beschriebenen Verfahren wurden nur Bereiche innerhalb der Gesichtsmaske verwendet.

[0243] Variante 5: Die Farbabstandsabbildung wird zuerst auf zwei Pegel quantisiert, sodass jeder Pixel entweder Hautfarbe (1) oder Nicht-Hautfarbe (–1) ist. Dies verhindert, dass die Amplitude der Farbabstandswerte einen übermäßigen Effekt auf den Abstand aus der Hautfarbenberechnung hat, d.h. bei Kombination mit der obigen Variante 4 haben stark nicht-hautfarbige Pixel außerhalb der Maske keinen übermäßigen Einfluss.

[0244] Variante 6: Die Hautfarbenmaske wird gradueller aktualisiert. Eine neue Maske wird als 90% der Maske des vorherigen Vollbildes berechnet, mit nur einer 10%igen Gewichtung der Maske des aktuellen Vollbildes. Dies hilft, die durch vorübergehende Teilabdeckungen, z.B. kurz vor dem Gesicht vorbeistreifende Hände, verursachten Probleme zu vermeiden. Es hilft auch, ein Problem von sich sehr langsam bewegenden Leuten zu vermeiden.

[0245] Die früher beschriebene Hautfarbennachführungstechnik arbeitete durch Abgleichen der Farbabstandsmarke des Gesichts im vorherigen Vollbild mit der Farbabstandsabbildung des aktuellen Vollbildes. Es gibt drei Aspekte dieses Verfahrens, welche wenigstens einige der Varianten verbessern wollen:

(i) Im Fall von Teilabdeckungen (zum Beispiel eine Hand vor dem Gesicht) ist es möglich, dass anstelle des Gesichts die Hand verfolgt wird.

(ii) Weil die Gesichter auf einem groben Raster (entsprechend der Gesichtsgröße) verfolgt werden, ist es für die Farbabstandsabbildung möglich, langsam von dem Gesicht zu „rutschen", falls sich das Gesicht langsam bewegt.

(iii) Weil nur hautfarbige Bereiche berücksichtigt werden, ist es möglich, dass ein guter Abgleich an einem einfarbigen Hautfarbenbereich gefunden wird.

[0246] Die Variante 6, bei welcher die Farbabstandabbildung für das Gesicht nur langsam aktualisiert wird, befasst sich mit den Aspekten (i) und (ii). Ebenso wie das Erlangen eines guten Abgleichs für Hautfarbenbereiche in der Gesichtsmaske werden in der Variante 5, die sich mit (iii) befasst, auch nicht-gesichtsfarbige Be-

reiche berücksichtigt.

[0247] Genauer wird für jedes Gesicht eine binäre Maske aus der Farbabstandsabbildung des vorherigen Vollbildes an der Stelle des Gesichts in diesem Rahmen berechnet. Jedes Element der Maske ist entweder „1", falls es kleiner als der mittlere Farbabstandswert für die gesamte Maske ist (d.h. es ist hautfarbig) oder sonst „0" (nicht-hautfarbig).

[0248] Eine Suche wird über das Suchfester mittels der binären Maske ausgeführt. An jedem Punkt in dem Suchfester wird ein Wert durch Aufsummieren aller Farbabstandswerte, wo die Maske „1" ist, und Subtrahieren aller Farbabstandswerte, wo die Maske „0" ist, berechnet. Der beste Abgleich wird als die Position genommen, an welcher die Maske den minimalen summierten Farbabstandswert ergibt.

[0249] Für jedes Gesicht in dem Bild wird unter Bezug auf Fig. 28 die aktuelle Farbabstandsabbildung (für das Vollbild n) durch Aufsummieren von 90% der alten Farbabstandsabbildung an der alten Gesichtsposition **1500** mit 10% der neuen Farbabstandsabbildung an der neuen Gesichtsposition **1510** aktualisiert.

[0250] Dies hat den Effekt des langsamen Veränderns der Farbabstandsmaske für jedes Gesicht. Somit ändert, falls das Gesicht teilweise verdeckt ist, der verdeckende Gegenstand die Farbabstandsmaske nur leicht, sodass der Abgleichvorgang anstelle des verdeckenden Gegenstandes wahrscheinlich noch das ursprüngliche Gesicht abgleicht.

[0251] Ein zusätzlicher Effekt dieses Verfahrens ist, dass ein sich langsam bewegendes Gesicht noch erfolgreich verfolgt wird, selbst wenn die Suche nur über ein grobes Raster ausgeführt wird. Der Grund hierfür ist offensichtlich, falls das folgende Szenario betrachtet wird:
Angenommen, die Suche findet mittels des früheren Verfahrens über ein 4×4-Raster statt (d.h. die Suche kann nicht genauer als 4 Pixel sein). Falls sich das Gesicht um ein Pixel bewegt, wird der Suchalgorithmus das Gesicht an der gleichen (vorherigen) Position finden. Da die Farbabstandsabbildung vollständig durch die Neue ersetzt wurde, wird, falls sich das Gesicht weiter mit einem Pixel je Vollbild bewegt, sich das Gesicht nach 4 Vollbildern noch an der ursprünglichen Position finden, obwohl es nun um 4 Pixel bewegt sein sollte – mit anderen Worten ist die Maske vom Gesicht „gerutscht".

[0252] Falls das Verfahren der Variante 6 verwendet wird, wo die Farbabstandsmaske langsam aktualisiert wird, wird der Suchalgorithmus nach 4 Vollbildern noch eine Farbabstandsbildung mit dem gleichen Speicher, wo das Gesicht 4 Vollbilder früher war, verwenden. Dies bedeutet, dass der Suchalgorithmus folgern kann, dass sich das Gesicht wirklich um 4 Pixel bewegt hat.

[0253] Eine weitere Verbesserung des Leitungsvermögens kann durch Quantisieren der Farbabstandsabbildung auf zwei Pegel beim Durchsuchen der Abbildung mit der binären Maske erzielt werden, wie bei Variante 5. Jedes Pixel in der Abbildung wird auf –1 gesetzt, falls es unter dem mittlere Farbabstandswert der Maske liegt, und auf +1, falls es über dem mittleren Farbabstandswert der Maske liegt (der „mittlere Farbabstand der Maske") bezieht sich auf den mittleren Farbabstand des Gesichtsfensters, bevor es auf „1" und „0" quantisiert wurde, um die Maske zu bilden.

[0254] Dann wird wie zuvor an jedem Punkt im Suchfenster ein Wert berechnet, in dem alle Farbabstandswerte, wo die Maske „1" ist, summiert werden und alle Farbabstandswerte, wo die Maske „0" ist, subtrahiert werden.

[0255] Zusammenfassend bedeutet dies, dass jede Pixelstelle im Gesichtsfenster basierend auf dem Maskenwert und dem Bildfarbabstandswert den folgenden Beitrag hat:

	Farbabstandsabbildungswert <= mittlerer Farbabstand der Maske (Hautfarben)	Farbabstandsabbildungswert > mittlerer Farbabstand der Maske (nicht-hautfarbig)
Pixel im Gesichtsfenster ist innerhalb der Maske (d.h. Maske = 1) (hautfarbig)	-1	+1
Pixel im Gesichtsfenster ist außerhalb der Maske (d.h. Maske = 0) (nicht-hautfarbig)	+1	-1

[0256] An jeder Position in dem Suchfenster werden diese Werte berechnet und für ein gesichtsgroßes Fenster summiert. Wie zuvor wird der beste Abgleich als die Pixelstelle genommen, wo der minimale Wert erzielt wird, d.h. wo die Maske am besten zum Bild passt.

[0257] Diese Verfahren verhindert, dass die Amplitude der Farbabstandswerte einen übermäßigen Effekt auf den Abstand von der Hautfarbenberechnung hat, sodass stark nicht-hautfarbige Pixel außerhalb der Maske keinen übermäßigen Einfluss haben.

Gesichtsnachführung und Dezimierung

[0258] In einem System mit Dezimierung, insbesondere zeitlicher Dezimierung wie oben beschrieben, kann es sein, dass es für einige Vollbilder kein definitives Gesichtserkennungsergebnis am Ende der Verarbeitung dieses Vollbildes gibt. In diesem Fall könnte die Hautfarbennachführung bezüglich der Vollbilder verwendet werden, für welche die zeitliche Dezimierung bedeutet, dass keine Gesichtserkennung verfügbar ist. Die Regeln über die Anzahl erforderlicher fortlaufender Gesichtserkennungen und dergleichen würde entsprechend gelockert werden. Falls eine räumliche Dezimierung bedeutet, dass ein Ergebnis bezüglich nur eines Teils eines Bildes verfügbar ist, könnte die Gesichtsnachführung immer durchgeführt werden, wenn ein komplett neues Ergebnis erzielt wurde, oder bei jedem Bild, auf der Basis der besten verfügbaren Informationen zu dieser Zeit bezüglich des gesamten Bildes.

Gradienten-Vorverarbeitung

[0259] Fig. 29a bis Fig. 29c zeigen schematisch eine Gradientenvorverarbeitungstechnik.

[0260] Es wurde bemerkt, dass Bildfenster mit einer geringen Pixelvariation dazu neigen, durch eine Gesichtserkennungsanordnung basierend auf Eigengesichtern oder Eigenblöcken als Gesichter erkannt zu werden. Deshalb wird ein Vorverarbeitungsschritt vorgeschlagen, um Bereiche geringer Pixelveränderung aus dem Gesichtserkennungsprozess zu entfernen. Im Fall eines Mehrmaßstabssystems (siehe oben) kann der Vorverarbeitungsschritt bei jedem Maßstab ausgeführt werden.

[0261] Der Basisprozess ist, dass ein „Gradiententest" auf jede mögliche Fensterposition über das gesamte Bild angewendet wird. Eine vorbestimmte Pixelposition für jede Fensterposition, wie beispielsweise das Pixel bei der oder am nächsten zur Mitte dieser Fensterposition, wird gekennzeichnet oder markiert in Abhängigkeit von den Ergebnissen des auf dieses Fenster angewendeten Tests. Falls der Test zeigt, dass ein Fenster eine geringe Pixelveränderung hat, wird diese Fensterposition in dem Gesichtserkennungsprozess nicht verwendet.

[0262] Ein erster Schritt ist in Fig. 29a veranschaulicht. Diese zeigt ein Fenster an einer beliebigen Fensterposition in dem Bild. Wie oben erwähnt, wird die Vorverarbeitung bei jeder möglichen Fensterposition wiederholt. Bezug nehmend auf Fig. 29a wurde festgestellt, obwohl die Gradientenvorverarbeitung auf das gesamte

Fenster angewendet werden könnte, dass bessere Ergebnisse erzielt werden, falls die Vorverarbeitung auf einen zentralen Bereich **1000** der Prüffensters **1010** angewendet wird.

[0263] Bezug nehmend auf Fig. 29b wird eine gradienten-basierte Messung aus dem Fenster (oder dem zentralen Bereich des Fensters, wie in Fig. 29a dargestellt) abgeleitet, welche der Mittelwert der absoluten Differenzen zwischen allen angrenzenden Pixeln **1011** in sowohl der horizontalen als auch der vertikalen Richtung über das Fenster ist. Jede Fenstermittenposition wird mit dieser gradierten-basierten Messung gekennzeichnet, um eine „Gradientenabbildung" des Bildes zu erzeugen. Die resultierende Gradientenabbildung wird dann mit einem Gradientenschwellenwert verglichen. Alle Fensterposition, für welche die gradienten-basierte Messung unter dem Gradientenschwellenwert liegt, werden von dem Gesichtserkennungsvorgang bezüglich dieses Bildes ausgeschlossen.

[0264] Alternative gradienten-basierte Messungen könnten verwendet werden, beispielsweise die Pixelvarianz oder der mittlere absolute Pixelunterschied von einem mittleren Pixelwert.

[0265] Die gradienten-basierte Messung wird vorzugsweise bezüglich Pixelluminanzwerten ausgeführt, aber kann natürlich auch auf andere Bildkomponenten eines Farbbildes angewendet werden.

[0266] Fig. 29c zeigt schematisch eine von einem Beispielbild abgeleitete Gradientenabbildung. Hier wird ein niedriger Gradientenbereich **1070** (schraffiert dargestellt) von der Gesichtserkennung ausgeschlossen, und nur ein höherer Gradientenbereich **1080** wird verwendet. Die oben beschriebenen Beispiele bezogen sich auf ein Gesichtserkennungssystem (einschließlich Trainings- und Erkennungsphasen) und mögliche Verwendungen davon in einem Kamerarecorder und einem Editiersystem. Es ist offensichtlich, dass es viele weitere mögliche Verwendungen solcher Techniken gibt, zum Beispiel (und nicht darauf beschränkt) Sicherheitsüberwachungssysteme, Medienbearbeitung allgemein (beispielsweise Videorecorder-Steuerungen), Videokonferenzsysteme, IP-Kameras, digitale Standkameras und dergleichen.

[0267] In anderen Ausführungsbeispielen können Fensterpositionen mit hohen Pixelunterschieden auch gekennzeichnet oder markiert werden und ebenfalls von dem Gesichtserkennungsprozess ausgeschlossen werden. Ein „hoher" Pixelunterschied bedeutet, dass die oben bezüglich Fig. 29b beschriebene Messung einen oberen Schwellenwert überschreitet.

[0268] So wird eine Gradientenabbildung wie oben beschreiben erzeugt. Alle Positionen, für welche die Gradientenmessung niedriger als der früher erwähnte (erste) Gradientenschwellenwert ist, werden von der Gesichtserkennungsverarbeitung ausgeschlossen, sowie alle Positionen, für welche die Gradientenmessung höher als der obere Schwellenwert ist.

[0269] Es wurde oben erwähnt, dass die Verarbeitung des „unteren Schwellenwerts" vorzugsweise auf einen zentralen Teil **1000** des Prüffensters **1010** angewendet wird. Gleiches kann auf die Verarbeitung des „oberen Schwellenwerts" angewendet werden. Dies würde bedeuten, dass nur eine einzelne Gradientenmessung bezüglich jeder Fensterposition abgeleitet werden muss. Alternativ kann, falls das gesamte Fenster bezüglich des unteren Schwellenwerttests verwendet wird, das gesamte Fenster in ähnlicher Weise bezüglich des oberen Schwellenwerttests verwendet werden. Wieder muss nur eine einzelne Gradientenmessung für jede Fensterposition abgeleitet werden. Es ist jedoch natürlich möglich, zwei verschiedene Anordnungen zu benutzen, sodass (zum Beispiel) ein zentraler Teil **1000** des Prüffensters **1010** verwendet wird, um die Gradientenmessung für den unteren Schwellenwerttest abzuleiten, aber das volle Prüffenster bezüglich des oberen Schwellenwerttests verwendet wird.

[0270] Ein weiteres Kriterium zum Zurückweisen einer Gesichtsverfolgung, wie früher erwähnt, besteht darin, dass seine Varianz oder Gradientenmessung sehr niedrig oder sehr hoch ist.

[0271] Bei dieser Technik wird eine verfolgte Gesichtsposition durch eine Varianz aus einem interessierenden Bereich der Abbildung validiert. Nur ein gesichtsgroßer Bereich der Abbildung bei dem erkannten Maßstab wird je Gesicht für die nächste Iteration der Nachführung gespeichert.

[0272] Trotz der oben beschriebenen Gradientenvorverarbeitung ist es für ein hautfarbenverfolgtes oder ein Kalman-vorhergesagtes Gesicht nach wie vor möglich, sich in einen (nicht-gesichtsartigen) niedrigen oder hohen Varianzbereich des Bildes zu bewegen. So werden während der Gradienten-Vorverarbeitung die Varianzwerte (oder Gradientenwerte) für die um Gesichtsnachführungen existierende Bereiche gespeichert.

[0273] Wenn die Endentscheidung über die nächste Position des Gesichts gemacht ist (mit irgendeinem Akzeptanztyp, entweder Gesichtserkennung, Hautfarben- oder Kalman-Vorhersage), wird die Position gegen die gespeicherten Varianz- (oder Gradierten-) Werte in dem interessierenden Bereich der Abbildung validiert. Falls die Position von sehr hoher oder sehr niedriger Varianz (oder Gradiertenwert) gefunden wird, wird sie als nicht-gesichtsartig erkannt und die Gesichtsnachführung wird beendet. Dies verhindert, dass Gesichtsnachführungen zu niedrigen (oder hohen) Varianzen Hintergrundbereichen des Bildes abwandern.

[0274] Alternativ kann, selbst wenn die Gadienten-Vorverarbeitung nicht verwendet wird, die Varianz der neuen Gesichtsposition frisch berechnet werden. In jedem Fall kann die verwendete Varianzmessung entweder eine herkömmliche Varianz oder die Summe von Differenzen benachbarter Pixel (Gradienten) oder irgendeine andere Messung des Varianztyps sein.

Kombinieren von Varianzvorverarbeitung und Änderungserkennung

[0275] Wie oben erwähnt, werden mit der Änderungserfassungstechnik ein oder mehrere rechtwinklige Begrenzungskästen um die Bereiche einer erkannten Bewegung platziert (oder zumindest um so Bereiche auszuschließen, die keine erkannte Bewegung haben). Diese Kästen werden dann auf alle Maßstäbe neu skaliert, bei denen eine Gesichtserkennung ausgeführt werden soll.

[0276] Die Entscheidung über den interessierenden Bereich, was sozusagen eine Entscheidung ist, welche Bereiche einer Gesichtserkennung unterzogen werden sollen, basiert auf den Ausgängen von den Varianzvorverarbeitungs- und Änderungserfassungsprozessen.

[0277] Nur Bereiche des Bildes, die
(i) als Gesichtsbereichskandidaten während der Varianzvorverarbeitung ausgewählt worden sind, und
(ii) als Bereiche einer erkannten Bewegung während der Änderungserfassung markiert worden sind,

müssen in dem Gesichtserkennungsalgorithmus verarbeitet werden.

[0278] Deshalb kombiniert die Entscheidungslogik des interessierenden Bereichs die interessierenden Bereiche aus den Varianzvorverarbeitungs- und Änderungserfassungsmodulen, um einen letzten interessierenden Bereich zu erzeugen. Diese werden durch einen oder mehrere rechtwinklige Begrenzungskästen bei jedem Maßstab oder (ohne Beschränkung auf Begrenzungskästen) einer Mehrmaßstabs-Abbildung „interessierender Bereiche" begrenzt, wobei jede Pixelposition als ein interessierender Bereich oder nicht markiert wird.

[0279] Es ist für den Gesichtserkennungsalgorithmus auch nützlich, zu wissen, welche der Bereiche, die nicht verarbeitet werden, einem „Mangel an erfasster Änderung" entsprechen, sodass die Gesichtserkennungsergebnisse aus dem vorherigen Vollbild für diese Bereiche wiederholt werden können. Deshalb wird auch diese Information zur Gesichtserkennung weitergegeben.

[0280] Es ist selbstverständlich, dass die oben beschriebenen Ausführungsbeispiele der Erfindung natürlich zumindest teilweise unter Verwendung von software-gesteuerten Datenverarbeitungsvorrichtungen verwirklicht werden können. Zum Beispiel können eine oder mehrere der schematisch dargestellten oder oben beschriebenen Komponenten als eine Software-gesteuerte Mehrzweck-Datenverarbeitungsvorrichtung oder ein maßgeschneidertes, programmgesteuertes Datenverarbeitungsgerät, wie beispielsweise eine anwendungsspezifische integrierte Schaltung, ein feldprogrammierbares Gate Array oder dergleichen realisiert werden. Es ist selbstverständlich, dass ein Computerprogramm, das eine solche Software bereitstellt, oder eine Programmsteuerung und ein Speicher-, Übertragung- oder anderes Bereitstellungsmedium, durch welches ein solches Computerprogramm gespeichert wird, als Aspekte der vorliegenden Erfindung ins Auge gefasst werden.

[0281] Es folgt die Liste der Literaturhinweise und Anhänge. Um Zweifel zu vermeiden, wird angemerkt, dass die Liste und die Anhänge einen Teil der vorliegenden Beschreibung bilden.

LITERATURHINWEISE

1. H. Schneiderman und T. Kanade, "A statistical model for 3D object detection applied to faces and cars", IEEE Conference on Computer Vision and Pattern Detection, 2000.
2. H. Schneiderman und T. Kanade, "Probabilistic modelling of local appearance and spatial relationships for object detection", IEEE Conference on Computer Vision and Pattern Detection, 1998.

3. H. Schneiderman, "A statistical approach to 3D object detection applied to face and cars", PhD thesis, Robotics Institute, Carnegie Mellon University, 2000.
4. E. Hjelmas und B.K. Low, "Face Detection: A Survey", Computer Vision and Image Understanding, Nr.83, Seiten 236–274, 2001.
5. M.-H. Yang, D. Kriegman und N. Ahuja, "Detecting Faces in Images: A Survey", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 24, Nr. 1, Seiten 34–58, Jan 2002.

ANHANG A: GESICHTSSÄTZE TRAINIEREN

[0282] Eine Datenbank besteht aus vielen tausend Bildern von Gegenständen, die vor einem Innenraum-Hintergrund stehen. Eine weitere Trainingsdatenbank, die in experimentellen Realisierungen der obigen Techniken verwendet wurde, besteht aus mehr als zehntausend 8-Bit-Graustufenbildern von menschlichen Köpfen mit Ansichten im Bereich von frontalen bis Links- und Rechtsprofilen. Der Fachmann versteht natürlich, dass verschiedene unterschiedliche Trainingssätze verwendet werden können, optional im Profil, um Gesichtseigenschaften einer lokalen Population zu reflektieren.

ANHANG B: EIGENBLÖCKE

[0283] Im Eigengesicht-Ansatz zur Gesichtserkennung (Literaturhinweise 4 und 5) wird jedes m×n-Gesichtsbild so aufgezeichnet, dass es durch einen Vektor einer Länge mit dargestellt wird. Jedes Bild kann dann als ein Punkt in dem mn-dimensionalen Raum angesehen werden. Ein Satz von Bildern wird zu einer Sammlung von Punkten in diesem großen Raum zu einer Abbildung gemacht.

[0284] Gesichtsbilder von ähnlichem Gesamtaufbau sind nicht zufällig in diesem mn-dimensionalen Bildraum verteilt und sie können deshalb durch einen relativ niederdimensionalen Subraum beschrieben werden. Eine Hauptkomponentenanalyse (PCA) benutzend können die Vektoren gefunden werden, die am besten auf die Verteilung der Gesichtsbilder in dem gesamten Bildraum entfallen. PCA involviert das Bestimmen der Haupteigenvektoren der Kovarianzmatrix entsprechend den ursprünglichen Gesichtsbildern. Diese Vektoren definieren den Subraum der Gesichtsbilder, häufig als face spuce bezeichnet. Jeder Vektor stellt ein m×n-Bild dar und ist eine Linearkombination der ursprünglichen Gesichtsbilder. Weil die Vektoren die Eigenvektoren der Kovarianzmatrix entsprechend den ursprünglichen Gesichtsbildern sind und weil sie von gesichtsartigem Aussehen sind, werden sie häufig als Eigengesichter bezeichnet [4].

[0285] Wenn ein unbekanntes Bild präsentiert wird, wird es in den Gesichtsraum projiziert. Auf diese Weise wird es in Einheiten einer gewichteten Summe von Eigengesichtern ausgedrückt.

[0286] In den vorliegenden Ausführungsbeispielen wird ein eng orientierter Ansatz verwendet, um so genannte „Eigenblöcke" oder Eigenvektoren betreffend Blöcke des Gesichtsbildes zu erzeugen und anzuwenden. Ein Raster der Blöcke wird auf das Gesichtsbild (im Trainingssatz) oder das Prüffenster (während der Erkennungsphase) angewendet und ein eigenvektor-basierter Prozess, sehr ähnlich dem Eigengesichtsprozess, wird an jeder Blockposition angewendet (oder in einem alternativen Ausführungsbeispiel zum Einsparen von Datenverarbeitung wird der Prozess einmal auf die Gruppe von Blockpositionen angewendet, wodurch ein Satz von Eigenblöcken zur Verwendung an jeder Blockposition erzeugt wird). Der Fachmann versteht, dass einige Blöcke, wie beispielsweise ein häufig ein Nasenmerkmal des Bildes darstellender zentraler Block, beim Entscheiden, ob ein Gesicht vorhanden ist, signifikanter sein können.

Berechnung von Eigenblöcken

[0287] Die Berechnung von Eigenblöcken involviert die folgenden Schritte:

(1) Ein Trainingssatz von N_T Bildern wird benutzt. Diese werden in Bildblöcke jeweils einer Größe von m × n aufgeteilt. Damit erhält man für jede Blockposition einen Satz von Bildblöcken, einen von dieser Position in jedem Bild:

$$\left\{I_0^t\right\}_{t=1}^{N_T}.$$

(2) Ein normierter Trainingssatz von Blöcken

$$\left\{I_0^t\right\}_{t=1}^{N_T}$$

wird wie folgt berechnet:

Jeder Bildblock I_0^t aus dem ursprünglichen Trainingssatz wird normiert, um einen Mittelwert von Null und eine L2-Norm von 1 zu haben, um einen jeweiligen normierten Bildblock I^t zu erzeugen. Für jeden Bildblock I_0^t, t = 1..N_T, gilt:

$$I' = \frac{I_0' - mean\_I_0'}{\left\| I_0' - mean\_I_0' \right\|}$$

mit $$mean\_I_0' = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I_0'[i,j], \text{ und}$$

$$\left\| I_0' - mean\_I_0' \right\| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(I_0'[i,j] - mean\_I_0'\right)^2}$$

$$\text{(d.h. der L2-Norm von } \left(I_0' - mean\_I_0'\right))$$

(3) Ein Trainingssatz von Vektoren

$$\left\{x'\right\}_{t=1}^{N_T}$$

wird durch lexikografisches Umordnen der Pixelelemente jedes Bildblocks I^t gebildet. Das heißt, jeder m×n-Bildblock I^T wird in einen Vektor x^t einer Länge N = mn aufgezeichnet.
(4) Der Satz von Ableitungsvektoren

$$D = \left\{x'\right\}_{t=1}^{N_T}$$

wird berechnet. D hat N Zeilen und N_T Spalten.
(5) Die Kovarianzmatrix Σ wird berechnet:

$\Sigma = DD^T$

Σ ist eine symmetrische Matrix der Größe N × N.
(7) Der gesamte Satz von Eigenvektoren P und Eigenwerten λ_i, i = 1, ... N, der Kovarianzmatrix Σ ist gegeben durch Auflösen von:

$\Lambda = P^T \Sigma P$

Hierbei ist Λ eine N×N-Diagonalmatrix mit den Eigenwerten λ_i entlang seiner Diagonalen (in der Reihenfolge der Größe), und P ist eine N×N-Matrix mit dem Satz von N Eigenvektoren jeweils der Länge N. Diese Zerlegung ist auch als Karhunen-Loeve-Transformation (KLT) bekannt.

[0288] Die Eigenvektoren können als ein Satz von Merkmalen angesehen werden, die zusammen die Veränderung zwischen den Blöcken der Gesichtsbilder charakterisieren. Sie bilden eine orthogonale Basis, durch welche jeder Bildblock dargestellt werden kann, d.h. grundsätzlich jedes Bild kann fehlerfrei durch eine gewichtete Summe der Eigenvektoren dargestellt werden.

[0289] Falls die Anzahl von Datenpunkten im Bildraum (die Anzahl von Trainingsbildern) kleiner als die Dimension des Raums ist (N_T < N), dann gibt es nur N_T aussagekräftige Eigenvektoren. Die übrigen Eigenvektoren haben Eigenwerte von Null zugeordnet. Daher werden, weil typischerweise N_T < N, alle Eigenwerte mit i > N_T zu Null.

[0290] Außerdem werden, weil die Bildblöcke im Trainingssatz im Gesamtaufbau ähnlich sind (sie sind alle abgeleitet von Gesichtern), nur einige der verbleibenden Eigenvektoren sehr starke Unterschiede zwischen den Bildblöcken kennzeichnen. Diese sind die Eigenvektoren mit den größten zugeordneten Eigenwerten. Die anderen übrigen Eigenvektoren mit kleineren zugeordneten Eigenwerten kennzeichnen nicht solche großen Unterschiede und sie sind daher nicht so nützlich zum Erkennen oder Unterscheiden zwischen Gesichtern.

[0291] Deshalb werden bei PCA nur die M Haupteigenvektoren mit den Eigenwerten größter Amplitude berücksichtigt, wobei M < N_T, d.h. eine Teil-KLT wird durchgeführt. Kurz gesagt extrahiert PCA einen niederdi-

mensionalen Subraum der KLT-Basis entsprechend den Eigenwerten der größten Werte.

[0292] Weil die Hauptkomponenten die stärksten Veränderungen zwischen den Gesichtsbildern beschreiben, können sie im Aussehen Teilen von Gesichtsblöcken ähneln und werden hier als Eigenblöcke bezeichnet. Der Begriff Eigenvektoren kann jedoch gleichermaßen verwendet werden.

Gesichtserkennung mittels Eigenblöcken

[0293] Die Ähnlichkeit eines unbekannten Gesichts zu einem Gesicht oder seine „Gesichtshaftigkeit", kann durch Bestimmen, wie gut das Bild durch den Gesichtsraum dargestellt wird, gemessen werden. Dieser Prozess wird auf einer blockweisen Basis ausgeführt, unter Verwendung des gleichen Rasters von Blöcken, wie sie im Trainingsprozess verwendet wurden.

[0294] Die erste Stufe dieses Prozesses enthält das Projizieren des Bildes in den Gesichtsraum

Projektion eines Bildes in den Gesichtsraum

[0295] Vor dem Projizieren eines Bildes in den Gesichtsraum werden viele der gleichen Vorverarbeitungsschritte an dem Bild durchgeführt, wie sie am Trainingssatz durchgeführt wurden:

(1) Ein Testbildblock I_0 der Größe m × n wird erzielt.

(2) Der ursprüngliche Testbildblock I_0 wird normiert, um einen Mittelwert von Null und eine L2-Norm von 1 zu haben, um den normierten Testbildblock I zu erzeugen:

$$I = \frac{I_0 - mean\_I_0}{\|I_0 - mean\_I_0\|}$$

mit $$mean\_I_0 = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} I_0[i,j], \text{ und}$$

$$\|I_0 - mean\_I_0\| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(I_0[i,j] - mean\_I_0)^2}$$

(d.h. der L2-Norm von $(I_0 - mean\_I_0)$)

(3) Die Ableitungsvektoren werden durch lexikografisches Umordnen der Pixelelemente des Bildes berechnet. Das Bild wird in einen Ableitungsvektor x^t der Länge N = mn umgeordnet.

Nach diesen Vorverarbeitungsschritten wird der Ableitungsvektor x mittels des folgenden einfachen Schritts in den Gesichtsraum projiziert:

(4) Die Projektion in den Gesichtsraum involviert das Transformieren des Ableitungsvektors x in seine Eigenblockkomponenten. Dies involviert eine einfache Multiplikation mit den M Haupteigenvektoren (den Eigenblöcken) P_i, i = 1, ... M. Jede Gewichtung y_i erhält man wie folgt:

$y_i = P_i^T x$

wobei P_i der i-te Eigenvektor ist.

[0296] Die Gewichtungen y_i, i = 1, ... M beschreiben den Beitrag jedes Eigenblocks beim Darstellen des eingegebenen Gesichtsblocks.

[0297] Blöcke von ähnlichem Aussehen haben ähnliche Sätze von Gewichtungen, während Blöcke eines unterschiedlichen Aussehens unterschiedliche Sätze von Gewichtungen haben. Deshalb werden die Gewichtungen hier als Merkmalsvektoren zum Klassifizieren der Gesichtsblöcke während der Gesichtserkennung verwendet.

Patentansprüche

1. Objekterkennungsvorrichtung (**10**, **100**) zum Erkennen von Objekten in einem Prüfbild, wobei die Vorrichtung aufweist:

eine Einrichtung (**230**) zum Vergleichen von Blöcken eines Prüffensters des Bildes mit Referenzdaten, die die

Anwesenheit eines Objekts anzeigen, um Indexpunktzahlen zu erzeugen, die einen Ähnlichkeitsgrad zwischen einem Bereich und den Referenzdaten angeben;
eine Einrichtung zum Speichern von Wahrscheinlichkeitsdaten entsprechend möglichen Werten der Indexpunktzahl und der Blockposition;
eine Einrichtung, die bezüglich eines aktuellen Blocks zum Zugreifen auf einen Wahrscheinlichkeitswert aus der Speichereinrichtung in Abhängigkeit von dieser Blockposition im Prüffenster und der bezüglich dieses Blocks erzeugten Indexpunktzahl ausgebildet ist; und
eine Einrichtung zum Kombinieren der Wahrscheinlichkeitswerte entsprechend den Blöcken in einem Prüffenster, um ein Ergebnis zu erzeugen, das die Wahrscheinlichkeit angibt, dass das Prüffenster ein Objekt enthält, und gekennzeichnet durch
eine Zugriffseinrichtung, die ausgebildet ist, um auf zwei oder mehr Wahrscheinlichkeitswerte bezüglich einer aktuellen Blockposition und Indexpunktzahl zuzugreifen, wobei die zwei oder mehr Wahrscheinlichkeitswerte sich auf verschiedene Objektorientierungen beziehen.

2. Vorrichtung nach Anspruch 1, bei welcher die Wahrscheinlichkeitswerte in der Speichereinrichtung nach Objektorientierung, Blockposition und dann nach Indexpunktzahl geordnet sind.

3. Vorrichtung nach Anspruch 1 oder 2, bei welcher die Kombinationseinrichtung ausgebildet ist, um die Wahrscheinlichkeitswerte betreffend einzelne Objektorientierungen zu kombinieren, um ein jeweiliges Ergebnis für jede Objektorientierung zu erzeugen.

4. Vorrichtung nach einem der vorherigen Ansprüche, bei welcher die Wahrscheinlichkeitswerte in der Speichereinrichtung nach Objektorientierung, dann nach Blockposition, dann nach Indexpunktzahl geordnet sind.

5. Vorrichtung nach einem der vorherigen Ansprüche, bei welcher die Objektorientierungen wenigstens eine Frontalorientierung und ein zu einer Seite gedrehtes Objekt aufweisen.

6. Vorrichtung nach Anspruch 5, bei welcher die Objektorientierungen wenigstens eine Frontalorientierung, ein zu einer Seite gedrehtes Objekt und ein zur anderen Seite gedrehtes Objekt aufweisen.

7. Vorrichtung nach Anspruch 5 oder Anspruch 6, bei welcher die Wahrscheinlichkeitswerte betreffend die Frontalorientierung in der Speichereinrichtung mit einer höheren Auflösung als die Wahrscheinlichkeitswerte betreffend die anderen Objektorientierungen gespeichert sind.

8. Vorrichtung nach Anspruch 7, bei welcher die Wahrscheinlichkeitswerte betreffend die Frontalorientierung in der Speichereinrichtung mit der doppelten Auflösung der Wahrscheinlichkeitswerte betreffend die anderen Objektorientierungen gespeichert sind.

9. Vorrichtung nach einem der vorherigen Ansprüche, bei welcher die Vergleichseinrichtung ausgebildet ist, um die Blöcke auf einen oder mehrere Bildeigenvektoren zu projizieren.

10. Vorrichtung nach einem der vorherigen Ansprüche, bei welcher die Zugriffseinrichtung einen Cache-Speicher zum Speichern von kürzlich zugegriffenen Wahrscheinlichkeitswerten und Wahrscheinlichkeitswerten, die in der Speicherreihenfolge nahe den kürzlich zugegriffenen Wahrscheinlichkeitswerten liegen, aufweist.

11. Vorrichtung nach einem der vorherigen Ansprüche, bei welcher die Objekte Gesichter sind.

12. Videokonferenzvorrichtung mit einer Vorrichtung nach einem der vorherigen Ansprüche.

13. Überwachungsvorrichtung mit einer Vorrichtung nach einem der Ansprüche 1 bis 11.

14. Kameraanordnung mit einer Vorrichtung nach einem der Ansprüche 1 bis 11.

15. Verfahren zum Erkennen von Objekten in einem Prüfbild, wobei das Verfahren die Schritte aufweist:
Vergleichen (**340**) von Blöcken eines Prüffensters des Bildes mit Referenzdaten, die die Anwesenheit eines Objekts angeben, um Indexpunktzahlen zu erzeugen, die einen Ähnlichkeitsgrad zwischen einem Bereich und den Referenzdaten angeben;
Speichern (**320**) von Wahrscheinlichkeitsdaten entsprechend möglichen Werten der Indexpunktzahl und der

Blockposition;
Zugreifen auf einen Wahrscheinlichkeitswert aus den gespeicherten Wahrscheinlichkeitsdaten in Abhängigkeit von einer aktuellen Blockposition im Prüffenster und der bezüglich dieses Blocks erzeugten Indexpunktzahl; und
Kombinieren der Wahrscheinlichkeitswerte entsprechend den Blöcken in einem Prüffenster, um ein Ergebnis zu erzeugen, das die Wahrscheinlichkeit angibt, dass das Prüffenster ein Objekt enthält,
und gekennzeichnet durch
Zugreifen auf zwei oder mehr Wahrscheinlichkeitswerte bezüglich einer aktuellen Blockposition und Indexpunktzahl, wobei die zwei oder mehr Wahrscheinlichkeitswerte sich auf verschiedene Objektorientierungen beziehen.

16. Computersoftware mit einem Programmcode zum Ausführen eines Verfahrens nach Anspruch 15.

17. Bereitstellungsmedium zum Bereitstellen eines Programmcodes nach Anspruch 16.

18. Medium nach Anspruch 17, wobei das Medium ein Speichermedium ist.

19. Medium nach Anspruch 18, wobei das Medium ein Übertragungsmedium ist.

Es folgen 21 Blatt Zeichnungen

Anhängende Zeichnungen

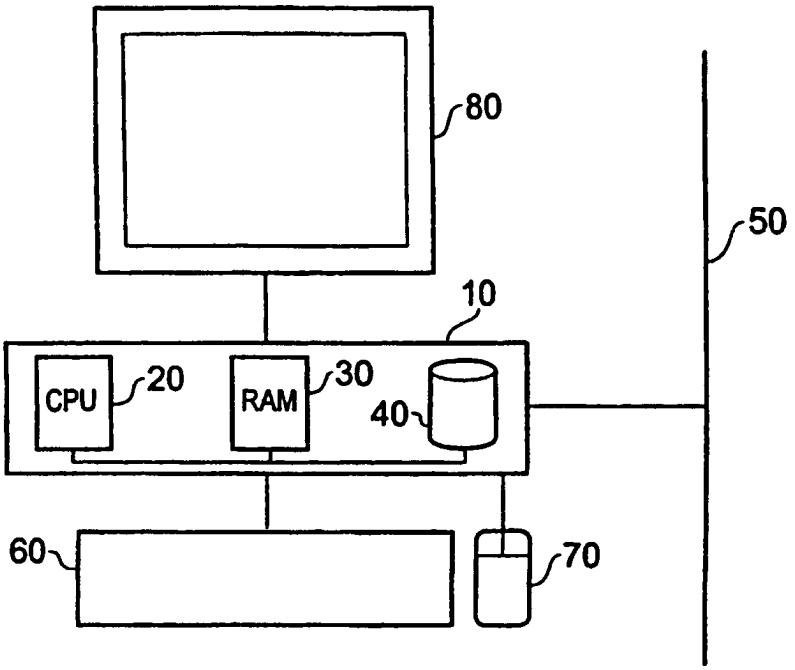


Fig. 1

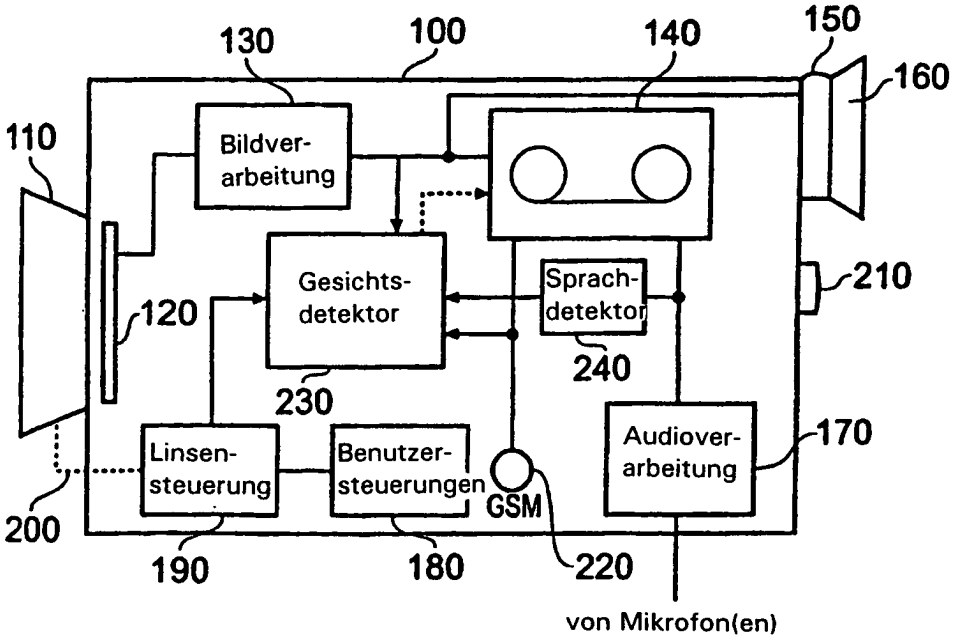


Fig. 2

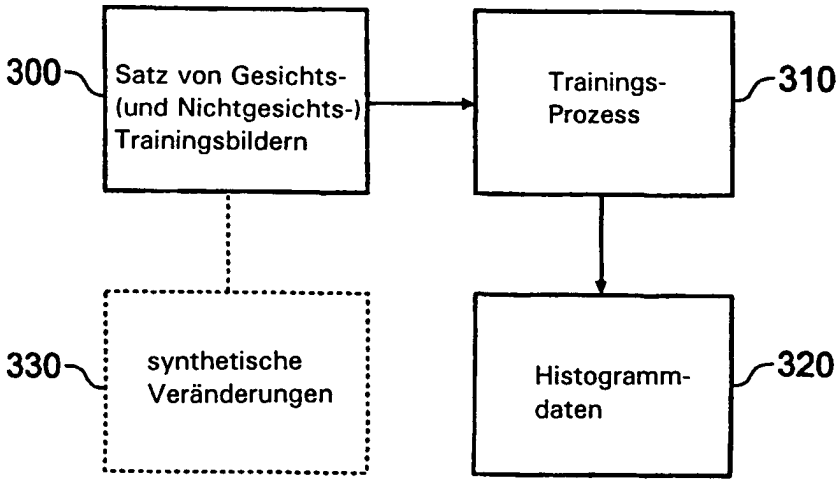

300
Satz von Gesichts- (und Nichtgesichts-) Trainingsbildern
Trainings- Prozess
310
330
synthetische Veränderungen
Histogramm- daten
320

Fig. 3

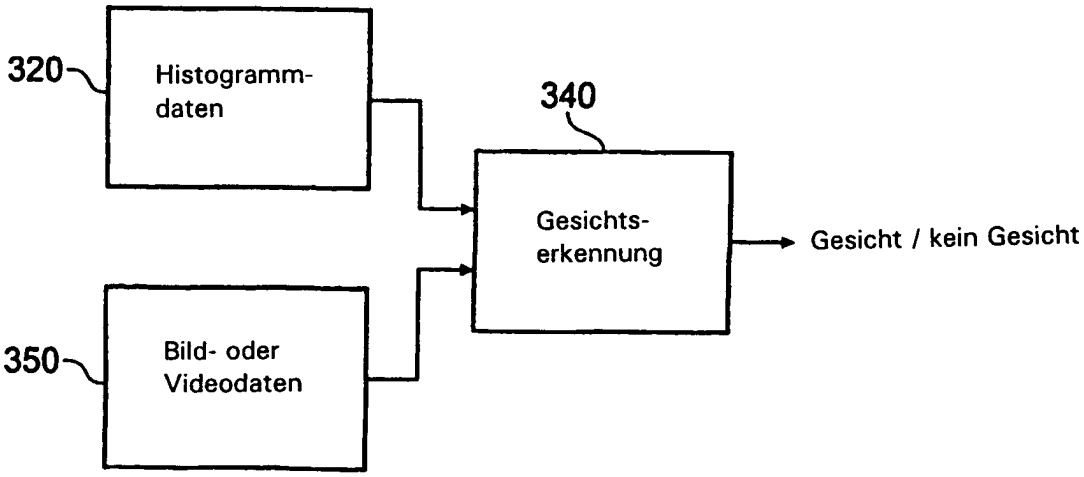

320
Histogramm- daten
340
Gesichts- erkennung
Gesicht / kein Gesicht
350
Bild- oder Videodaten

Fig. 4

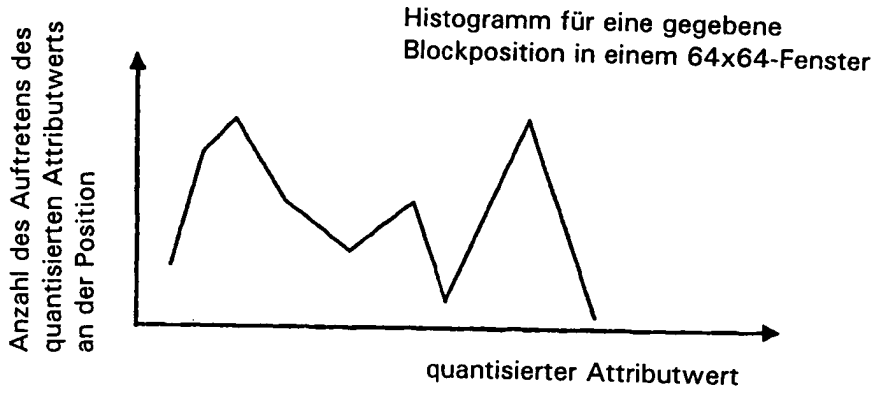

Histogramm für eine gegebene
Blockposition in einem 64x64-Fenster
Anzahl des Auftretens des
quantisierten Attributwerts
an der Position
quantisierter Attributwert

Fig. 5

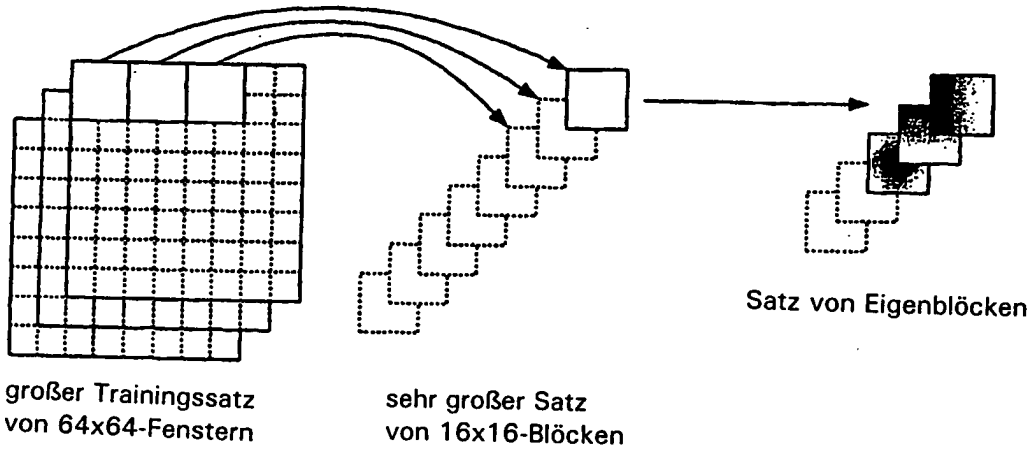

großer Trainingssatz
von 64x64-Fenstern
sehr großer Satz
von 16x16-Blöcken
Satz von Eigenblöcken

Fig. 6

Fig. 7

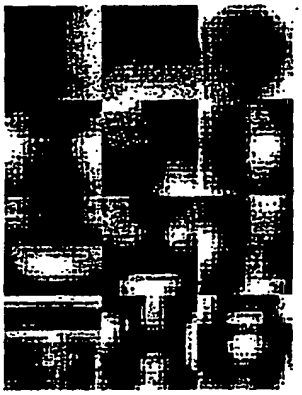


Fig. 8

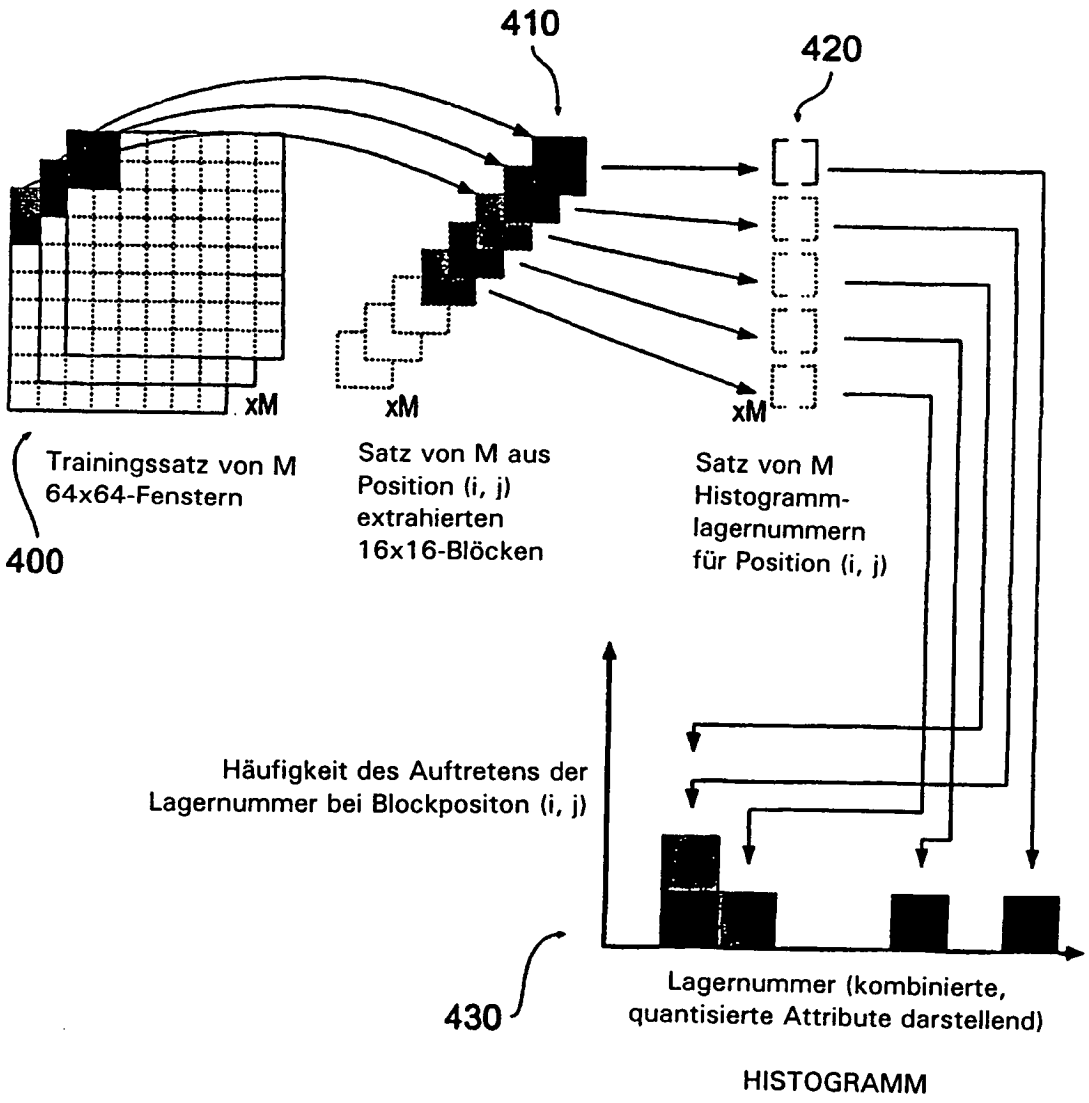

410
420
xM
xM
xM
Trainingssatz von M
64x64-Fenstern
400
Satz von M aus
Position (i, j)
extrahierten
16x16-Blöcken
Satz von M
Histogramm-
lagernummern
für Position (i, j)
Häufigkeit des Auftretens der
Lagernummer bei Blockpositon (i, j)
430
Lagernummer (kombinierte,
quantisierte Attribute darstellend)
HISTOGRAMM

Fig. 9

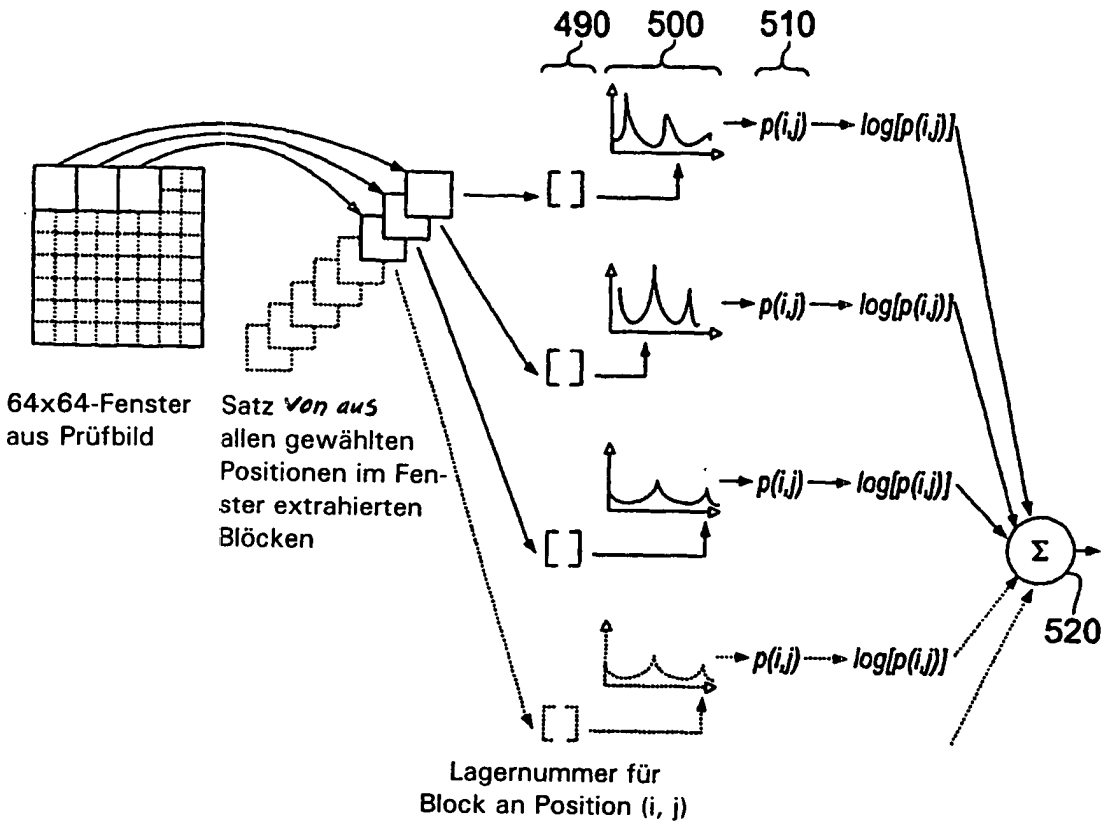

490
500
510
p(i,j)
log[p(i,j)]
64x64-Fenster aus Prüfbild
Satz von aus allen gewählten Positionen im Fenster extrahierten Blöcken
Σ
520
Lagernummer für Block an Position (i, j)

Fig. 11

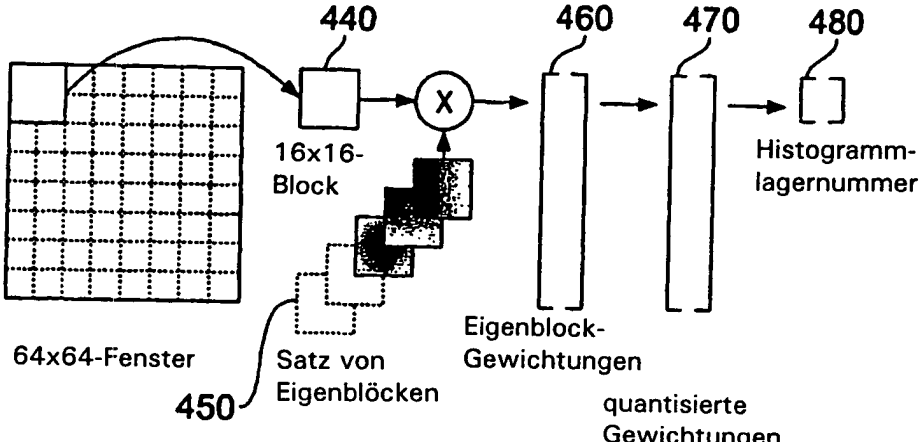

440
460
470
480
X
16x16-Block
Histogramm-lagernummer
64x64-Fenster
Satz von Eigenblöcken
450
Eigenblock-Gewichtungen
quantisierte Gewichtungen

Fig. 10

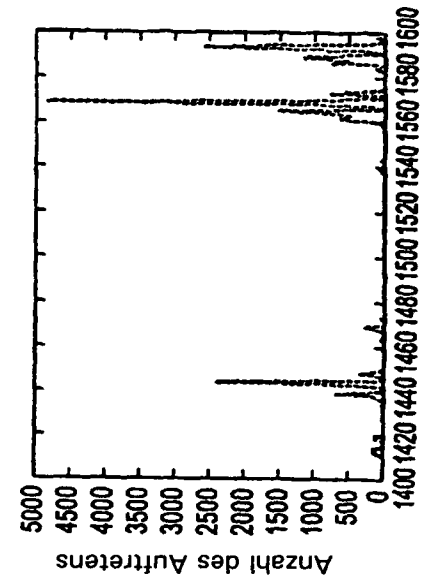


Fig. 12b

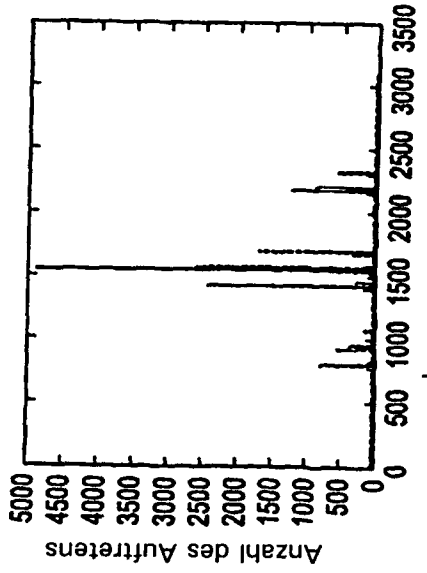


Fig. 12a

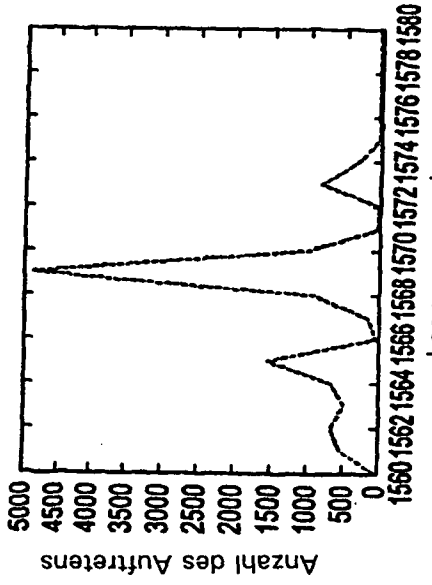


Fig. 12c

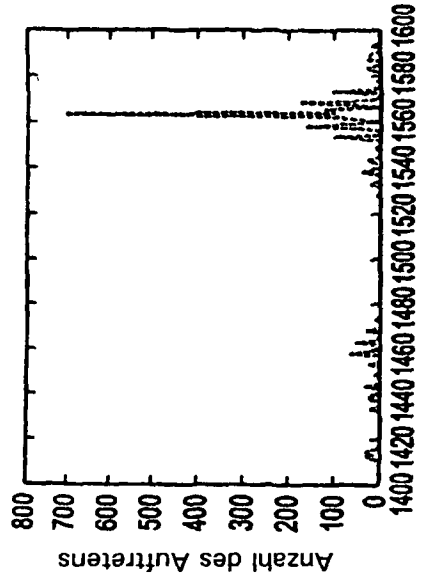


Fig. 12e

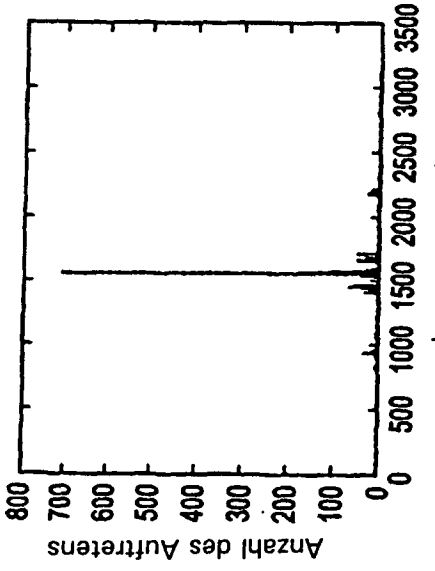


Fig. 12d

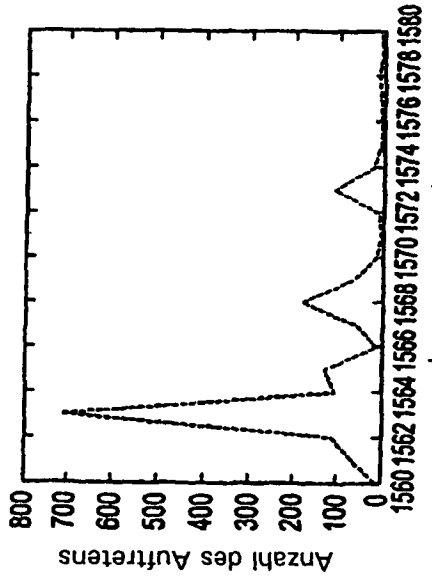


Fig. 12f

$F^{38}_{0,0,0}$	$F^{38}_{1,0,0}$	$F^{38}_{2,0,0}$	...	$F^{38}_{0,1,0}$	$F^{38}_{1,1,0}$	$F^{38}_{2,1,0}$	...	$F^{38}_{0,2,0}$	$F^{38}_{1,2,0}$	$F^{38}_{2,2,0}$	...	$F^{38}_{0,0,1}$
$F^{38}_{1,0,1}$	$F^{38}_{2,0,1}$	...	$L^{38}_{0,0,0}$	$L^{38}_{1,0,0}$	$L^{38}_{2,0,0}$	...	$R^{38}_{0,0,0}$	$R^{38}_{1,0,0}$	$R^{38}_{2,0,0}$	...	$F^{22}_{0,0,0}$	$F^{22}_{1,0,0}$
$F^{22}_{2,0,0}$												

Fig. 13a

$F^{38}_{0,0,0}$	$L^{38}_{0,0,0}$	$R^{38}_{0,0,0}$	$F^{38}_{0,1,0}$	$L^{38}_{0,1,0}$	$R^{38}_{0,1,0}$	$F^{38}_{0,2,0}$	$L^{38}_{0,2,0}$	$R^{38}_{0,2,0}$	...	$F^{38}_{0,0,1}$	$L^{38}_{0,0,1}$	$R^{38}_{0,0,1}$
$F^{38}_{1,0,0}$	$L^{38}_{1,0,0}$	$R^{38}_{1,0,0}$	...	$F^{22}_{0,0,0}$	$L^{22}_{0,0,0}$	$R^{22}_{0,0,0}$						

Fig. 13b

020	017	213	413	511	500	001	056	078	034	067	123	413	511	500	078
145	322	421	001	006	432	491	291	328	348	428	367	001	006	432	328
327	456	321	094	329	211	288	370	509	208	218	213	094	329	211	509
127	123	124	111	098	345	321	312	132	326	218	239	217	239	129	374
165	456	789	094	654	211	211	370	456	208	510	213	008	329	211	509
199	200	212	199	398	433	421	444	501	109	178	190	123	176	433	431
398	297	157	174	145	193	283	398	425	511	111	176	189	148	190	312
396	345	276	222	111	333	254	276	287	298	243	254	265	165	176	187
198	212	223	234	245	256	267	143	144	143	142	146	067	034	089	056
099	045	276	478	287	265	445	183	129	348	428	367	001	006	432	328
075	455	299	288	233	243	343	454	500	409	308	201	098	036	082	182
198	129	138	174	184	174	187	146	176	328	378	345	274	290	289	276
156	478	510	439	472	389	267	385	280	291	111	122	133	311	477	487
287	387	027	023	024	021	023	026	028	498	456	423	476	435	507	389
326	328	329	317	324	423	452	112	221	734	289	500	499	497	490	508

Fig. 14

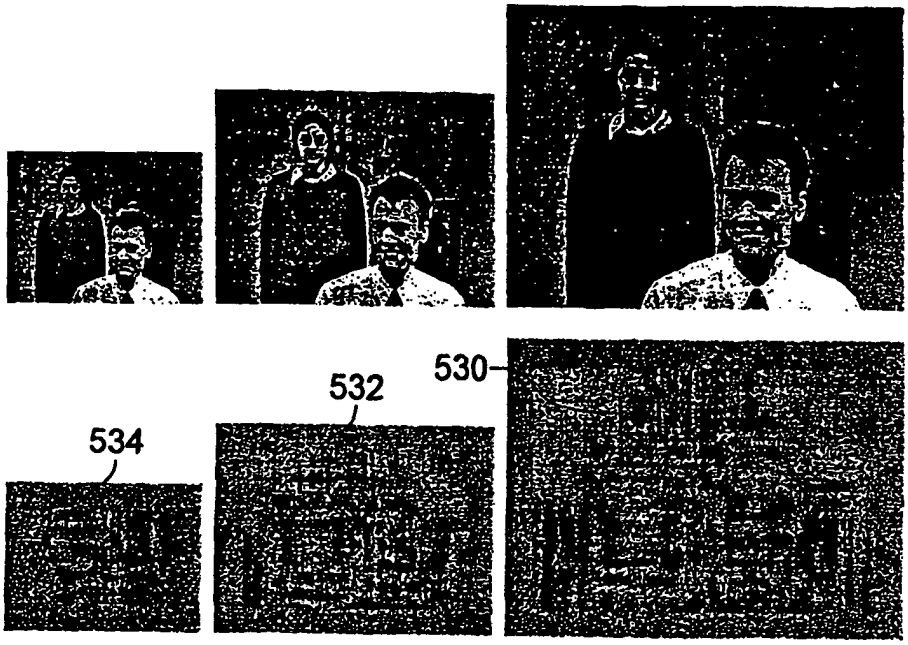


Fig. 15a Fig. 15b Fig. 15c

Fig. 15d

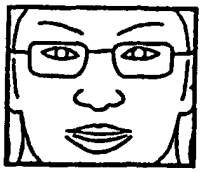


Fig. 15e

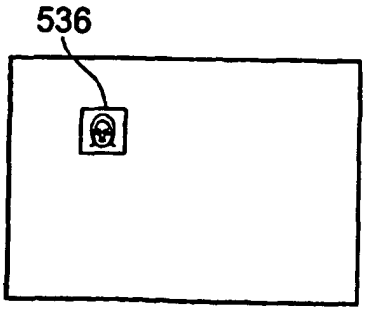


Fig. 15f

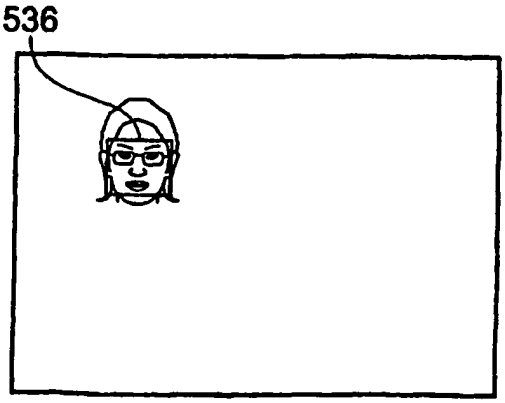


Fig. 15g

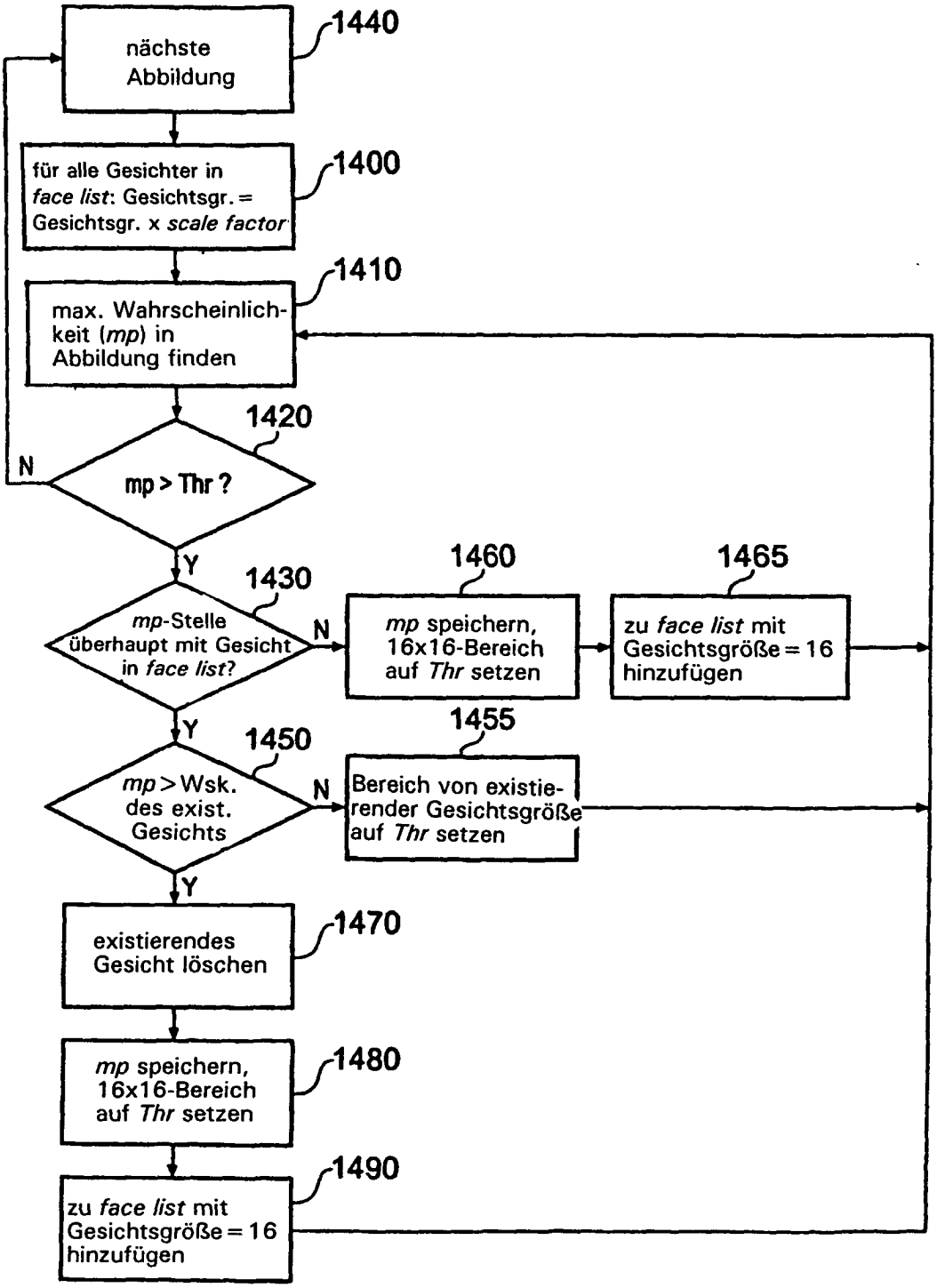

nächste Abbildung
1440
für alle Gesichter in face list: Gesichtsgr. = Gesichtsgr. x scale factor
1400
max. Wahrscheinlichkeit (mp) in Abbildung finden
1410
1420
mp > Thr ?
N
Y
1430
mp-Stelle überhaupt mit Gesicht in face list?
N
Y
1460
mp speichern, 16x16-Bereich auf Thr setzen
1465
zu face list mit Gesichtsgröße = 16 hinzufügen
1450
mp > Wsk. des exist. Gesichts
N
Y
1455
Bereich von existierender Gesichtsgröße auf Thr setzen
existierendes Gesicht löschen
1470
mp speichern, 16x16-Bereich auf Thr setzen
1480
zu face list mit Gesichtsgröße = 16 hinzufügen
1490

Fig. 16

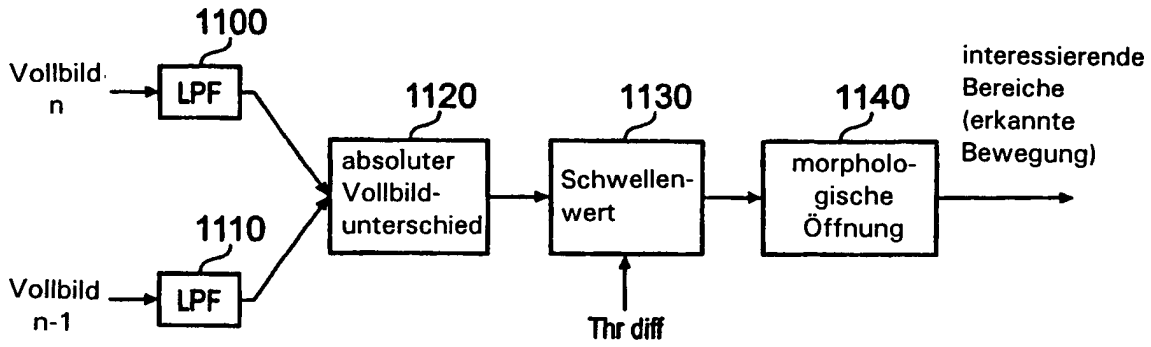

1100
Vollbild n
LPF
1110
Vollbild n-1
LPF
1120
absoluter Vollbild-unterschied
1130
Schwellen-wert
Thr diff
1140
morpholo-gische Öffnung
interessierende Bereiche (erkannte Bewegung)

Fig. 17

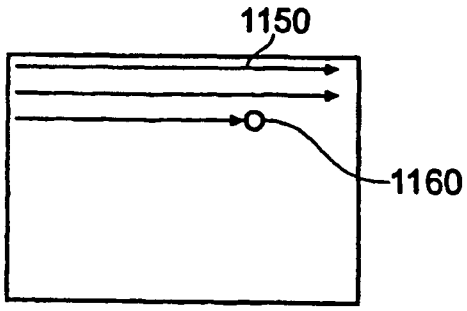


Fig. 18a

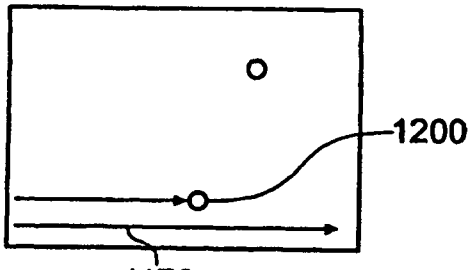


Fig. 18b

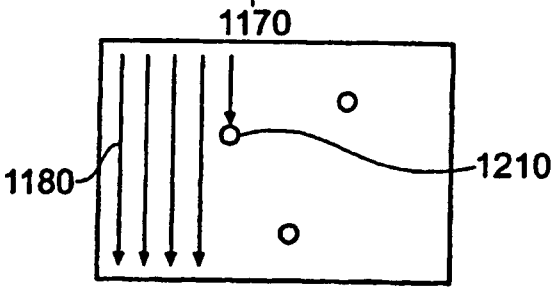


Fig. 18c

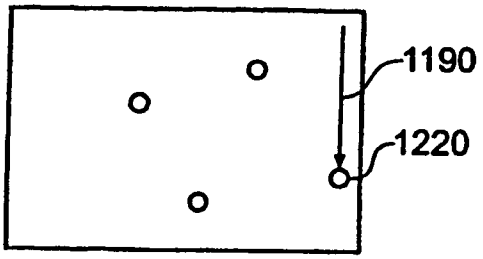


Fig. 18d

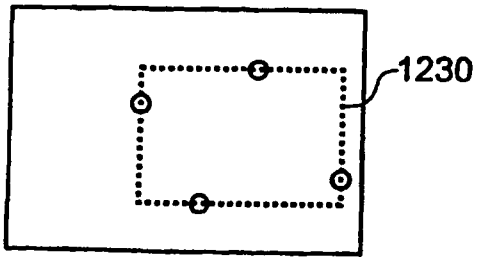


Fig. 18e

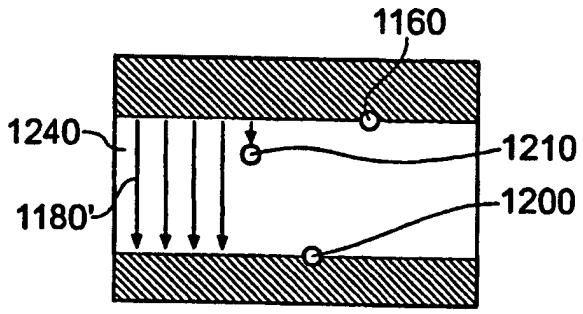


Fig. 19a

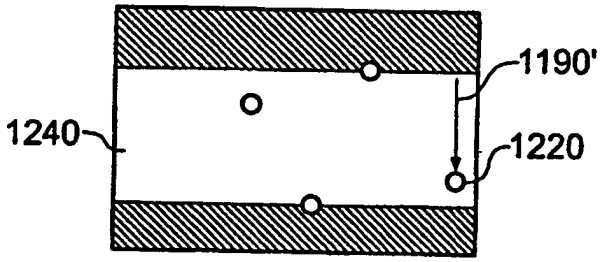


Fig. 19b

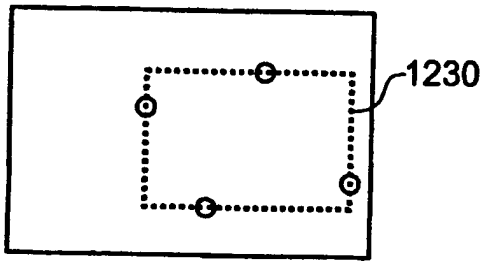
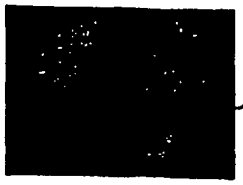


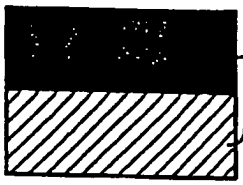
Fig. 19c

Fig. 20a



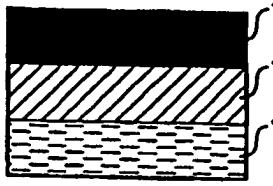
Raumfaktor 1 (d.h. keiner)

Fig. 20b

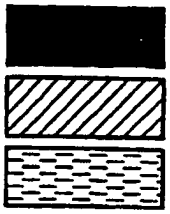


Raumfaktor 2

Fig. 20c



Raumfaktor 3

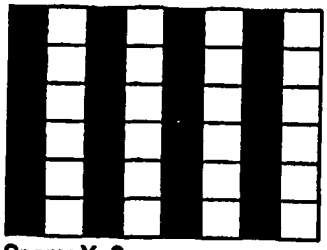


Bereiche, benutzt für Gesichtserkennungsvollbild n

Bereiche, benutzt für Gesichtserkennungsvollbild n + 1

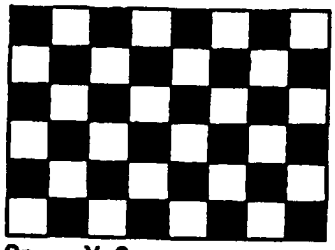
Bereiche, benutzt für Gesichtserkennungsvollbild n + 2

Fig. 21a



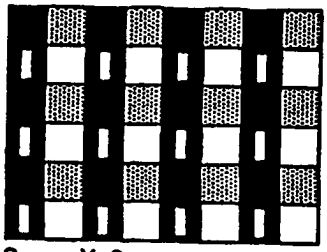
SparseX=2
SparseY=1
UseChequerBoard=0

Fig. 21b



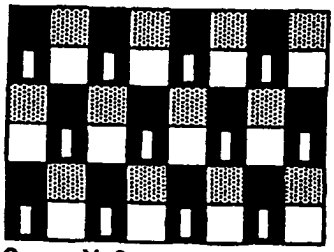
SparseX=2
SparseY=1
UseChequerBoard=1

Fig. 21c



SparseX=2
SparseY=2
UseChequerBoard=0

Fig. 21d



SparseX=2
SparseY=2
UseChequerBoard=1

Pixel, benutzt für Gesichtserkennungsvollbild n

Pixel, benutzt für Gesichtserkennungsvollbild n + 1

Pixel, benutzt für Gesichtserkennungsvollbild n + 2

Pixel, benutzt für Gesichtserkennungsvollbild n + 3

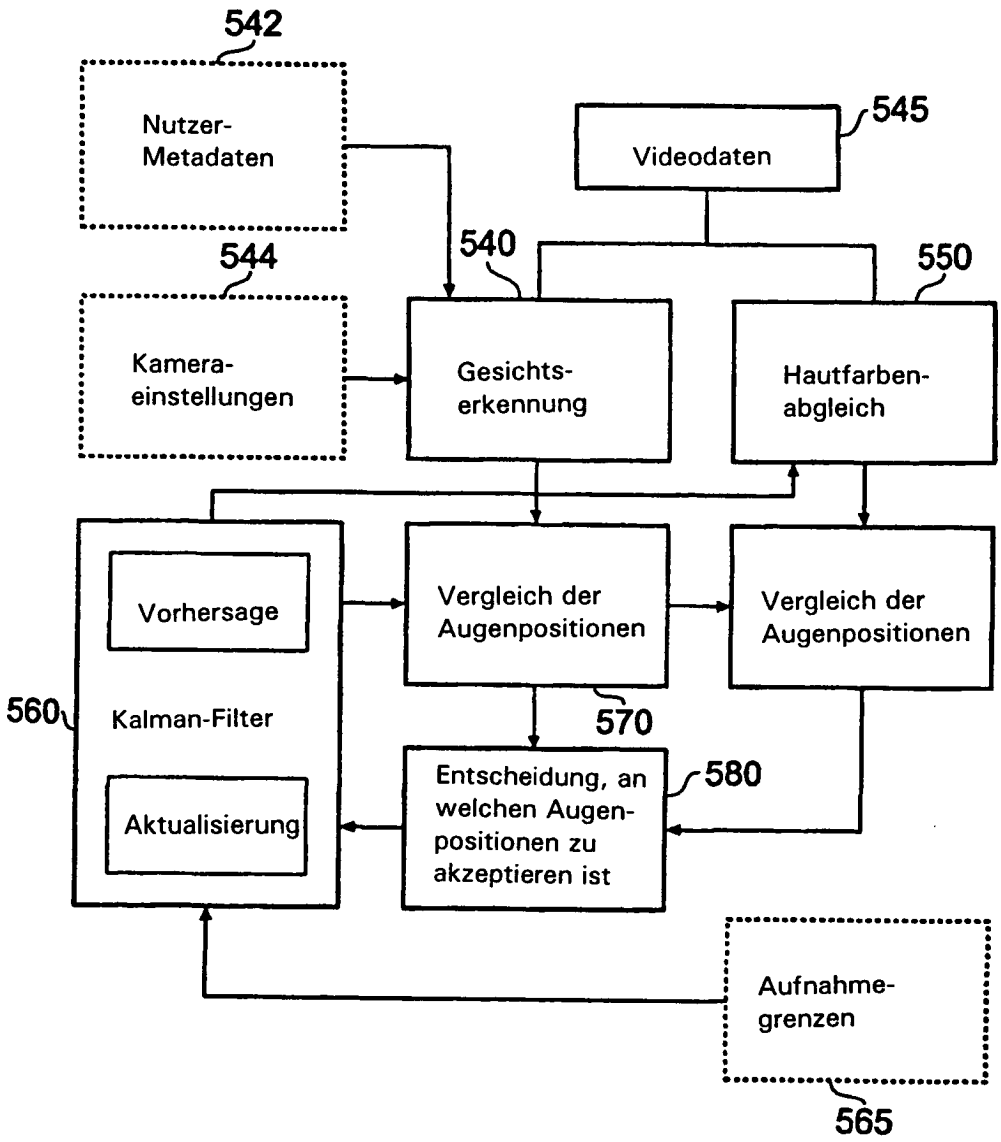

542
Nutzer-
Metadaten
545
Videodaten
544
540
550
Kamera-
einstellungen
Gesichts-
erkennung
Hautfarben-
abgleich
Vorhersage
Vergleich der
Augenpositionen
Vergleich der
Augenpositionen
560
Kalman-Filter
570
580
Entscheidung, an
welchen Augen-
positionen zu
akzeptieren ist
Aktualisierung
Aufnahme-
grenzen
565

Fig. 22

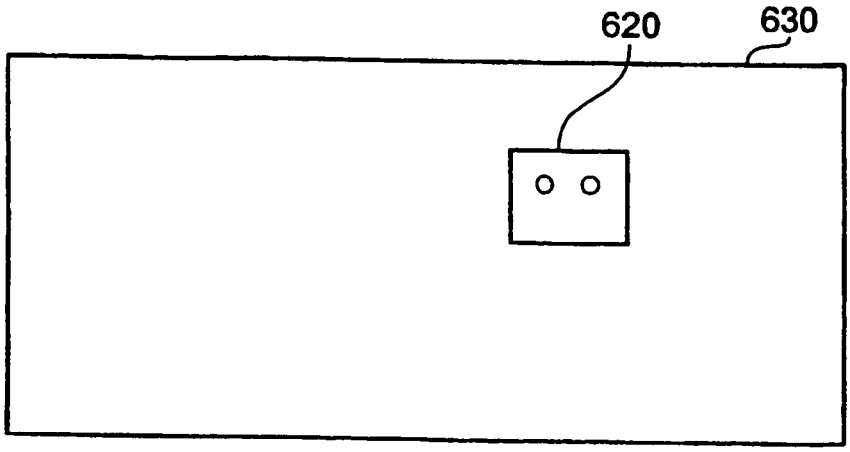


Fig. 23a

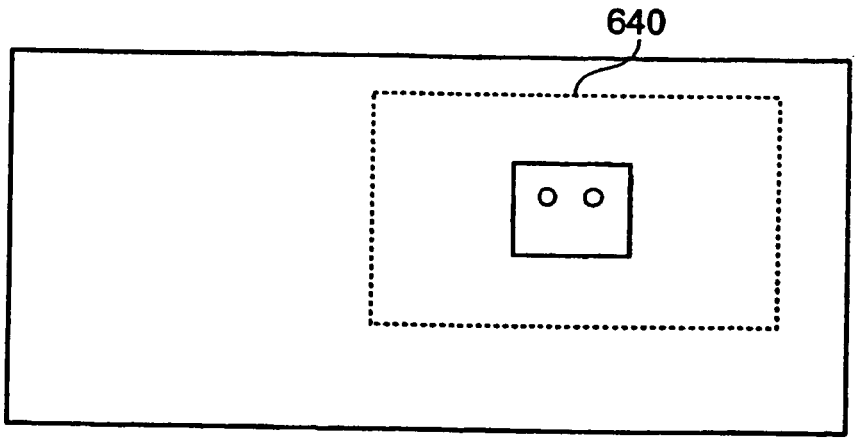


Fig. 23b

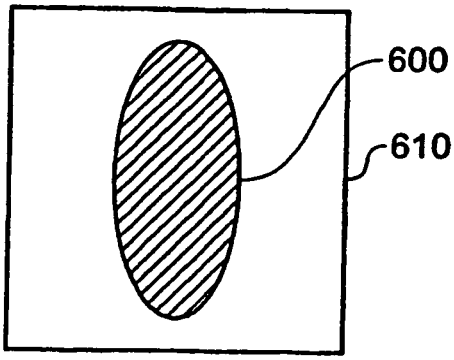


Fig. 24

Fig. 25a

Fig. 25b

Fig. 25c

Fig. 26

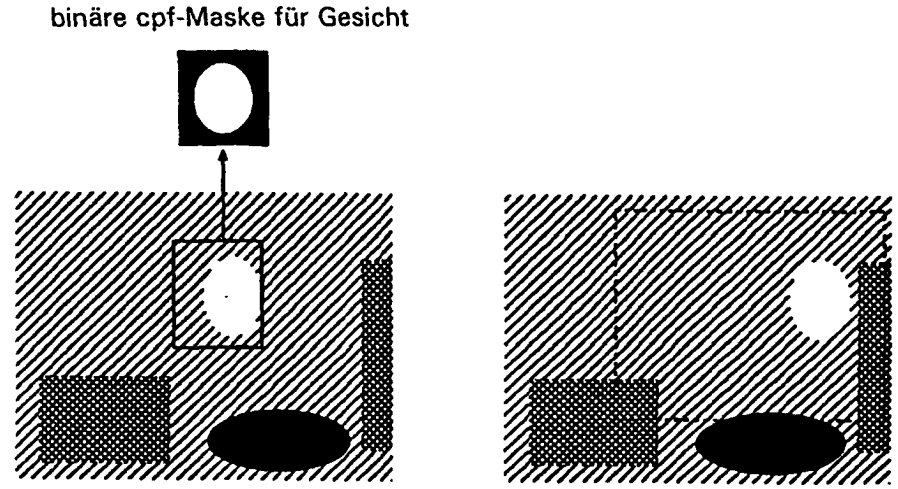

binäre cpf-Maske für Gesicht
cpf-Abbildung für Vollbild n-1
cpf-Abbildung für Vollbild n,
mit gestricheltem Suchfenster

Fig. 27

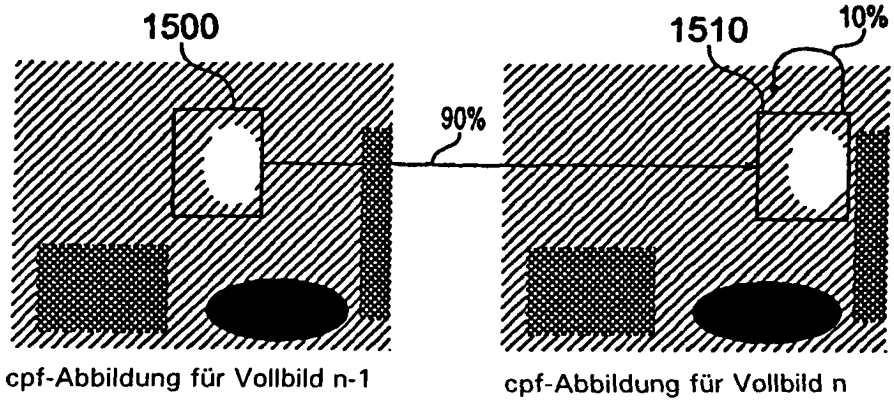

1500
1510
10%
90%
cpf-Abbildung für Vollbild n-1
cpf-Abbildung für Vollbild n

Fig. 28

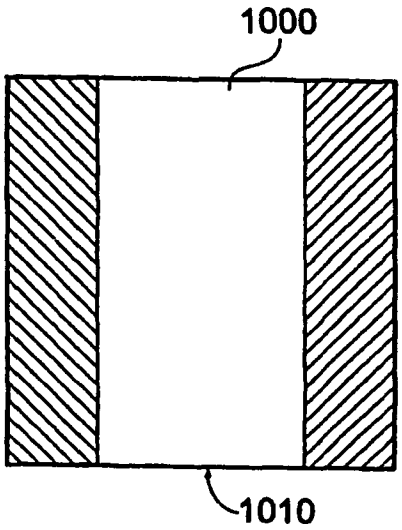


Fig. 29a

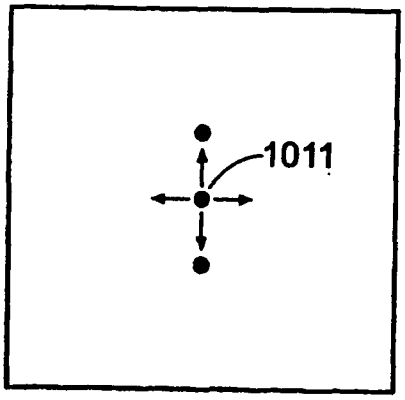


Fig. 29b

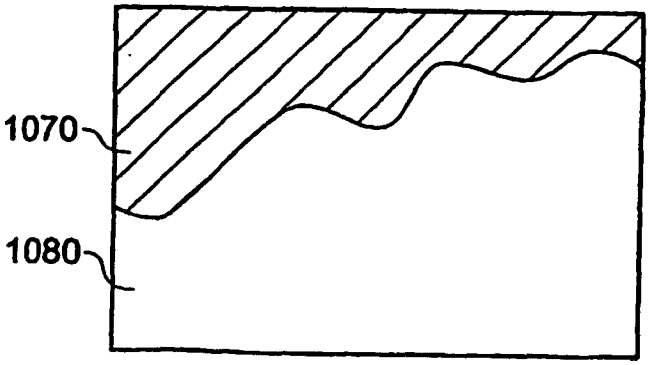


Fig. 29c