



(12) **Veröffentlichung**

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2020/140223**
in der deutschen Übersetzung (Art. III § 8 Abs. 2
IntPatÜG)
(21) Deutsches Aktenzeichen: **11 2019 006 564.5**
(86) PCT-Aktenzeichen: **PCT/CN2019/070174**
(86) PCT-Anmeldetag: **03.01.2019**
(87) PCT-Veröffentlichungstag: **09.07.2020**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **14.10.2021**

(51) Int Cl.: **G06K 9/66 (2006.01)**

(71) Anmelder:
Intel Corporation, Santa Clara, CA, US

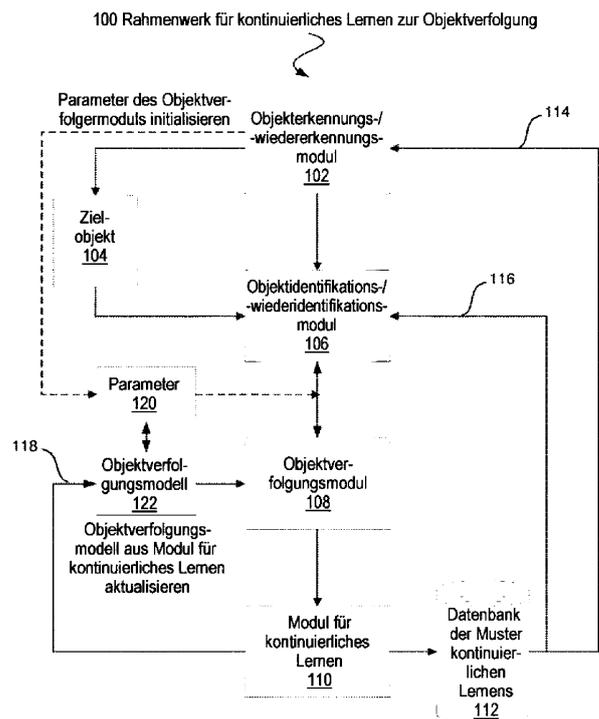
(72) Erfinder:
**Zhang, Lidan, Beijing, CN; Guo, Ping, Beijing, CN;
Ren, Haibing, Beijing, CN; Zhang, Yimin, Beijing,
CN**

(74) Vertreter:
HGF Europe LLP, 81673 München, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Kontinuierliches Lernen zur Objektverfolgung**

(57) Zusammenfassung: Ein Langzeit-Objektverfolger setzt ein Rahmenwerk für kontinuierliches Lernen ein, um Wanderung in der Verfolgungsposition eines verfolgten Objektes zu beseitigen. Das Rahmenwerk für kontinuierliches Lernen besteht aus einem Modul für kontinuierliches Lernen, das Muster des verfolgten Objektes akkumuliert, um die Genauigkeit der Objektverfolgung über verlängerte Zeiträume zu verbessern. Das Modul für kontinuierliches Lernen kann einen Musterpräprozessor, um einen Ort eines Kandidatenobjektes zu verfeinern, das während der Objektverfolgung gefunden wurde, und einen Ausschneider beinhalten, um einen Teil eines Einzelbildes, der ein verfolgtes Objekt enthält, als ein Muster auszuschneiden und das Muster in eine Datenbank für kontinuierliches Lernen einzufügen, um zukünftige Verfolgung zu unterstützen.



Beschreibung

TECHNISCHES GEBIET

[0001] Das technische Gebiet betrifft im Allgemeinen Computer-Vision und im Besonderen Objektverfolgung bei Computer-Vision.

ALLGEMEINER STAND DER TECHNIK

[0002] In der Computer-Vision bezeichnet Objektverfolgung Lokalisierung eines beliebigen Zieles in mehreren Einzelbildern einer Videosequenz oder in einem live erfassten Stream. Objektverfolgung wird in vielen Computer-Vision-Anwendungen verwendet, wie z.B. Videoanalyse, Überwachung, Robotik, Augmented Reality und Videobearbeitung. Eines der Ziele bei der Objektverfolgung ist es, ein Zielobjekt über einen längeren Zeitraum zu verfolgen, was hierin als Langzeit-Objektverfolgung bezeichnet wird. Aufgrund verschiedener Faktoren, die mit der Zeit Veränderungen im Erscheinungsbild des Zielobjektes bewirken, kann Langzeit-Objektverfolgung problematisch sein, wie z.B. Abdeckung durch andere Objekte, starre oder nicht starre Veränderung des Zielobjektes selbst, Kamerabewegung während des Aufnehmens von Bildern des Zielobjektes, die Bewegungsunschärfe verursachen kann, Beleuchtungsänderung usw. Beispielsweise kann sich, wenn das Zielobjekt eine Person ist, die unter Überwachung steht, während sie durch ein Flughafenterminal geht, das Erscheinungsbild der Person abhängig von ihrer Kleidung, ihrer Umgebung, ihren Handlungen und ihrer Position relativ zu anderen Personen in der Nähe, die durch das Terminal gehen, mit der Zeit ändern.

Figurenliste

[0003] Die beschriebenen Ausführungsformen sind in den Figuren der beiliegenden Zeichnungen als Beispiel und nicht einschränkend dargestellt. Die Verfahren, die Prozesse und die Logik, die in den nachstehenden Figuren abgebildet sind, können Hardware (z.B. Schaltungen, zweckbestimmte Logik, Steuerungen usw.), Software (wie sie z.B. auf einem Mehrzweck-Computersystem oder einer zweckbestimmten Maschine, z.B. einem Softwaremodul oder Logik, ausgeführt wird) und Schnittstellen (wie z.B. eine Speicherschnittstelle) zwischen Hardware und Software oder eine Kombination beider umfassen. Obgleich die abgebildeten Verfahren, Prozesse und die abgebildete Logik im Hinblick auf sequenzielle Operationen beschrieben sein können, muss man verstehen, dass einige der beschriebenen Operationen in unterschiedlicher Reihenfolge durchgeführt werden können. Darüber hinaus können einige Operationen parallel statt sequenziell ausgeführt werden. Die nachstehenden Figuren beinhalten ähnliche Bezugszeichen, die ähnliche Elemente angeben, und wobei

Fig. 1 eine schematische Blockdiagrammdarstellung eines Rahmenwerkes für kontinuierliches Lernen zur Langzeit-Objektverfolgung gemäß verschiedenen hierin beschriebenen Beispielen ist;

Fig. 2 eine schematische Blockdiagrammdarstellung einer Architektur zum Implementieren kontinuierlichen Lernens zur Langzeit-Objektverfolgung gemäß verschiedenen hierin beschriebenen Beispielen ist;

Fig. 3 eine schematische Blockdiagrammdarstellung einer Übersicht eines Moduls für kontinuierliches Lernen zum Implementieren kontinuierlichen Lernens zur Langzeit-Objektverfolgung gemäß verschiedenen hierin beschriebenen Beispielen ist;

Fig. 4 eine Darstellung einer Sequenz von Videoeinzelbildern ist, die ein Zielobjekt enthalten, für das kontinuierliches Lernen zur Langzeit-Objektverfolgung gemäß verschiedenen hierin beschriebenen Beispielen implementiert ist;

Fig. 5A-5B schematische Blockdiagramme sind, die kontrastierende Schemata für das Trainieren für Langzeit-Objektverfolgung mit kontinuierlichem Lernen und ohne dieses gemäß verschiedenen hierin beschriebenen Beispielen darstellen;

Fig. 6A-6E eine andere Sequenz von Videoeinzelbildern darstellt, die ein Zielobjekt enthalten, für das kontinuierliches Lernen zur Langzeit-Objektverfolgung gemäß verschiedenen hierin beschriebenen Beispielen implementiert ist;

Fig. 7A-7D eine weitere Sequenz von Videoeinzelbildern darstellt, die ein Zielobjekt enthalten, für das kontinuierliches Lernen zur Langzeit-Objektverfolgung gemäß verschiedenen hierin beschriebenen Beispielen implementiert ist; und

Fig. 8 eine schematische Blockdiagrammdarstellung eines typischen Computersystems ist, in dem Ausführungsformen kontinuierlichen Lernens zur Langzeit-Objektverfolgung entweder ganz oder zum Teil gemäß verschiedenen hierin beschriebenen Beispielen implementiert sein können.

[0004] Andere Merkmale der beschriebenen Ausführungsformen werden aus den beiliegenden Zeichnungen und aus der nachstehenden ausführlichen Beschreibung offensichtlich.

BESCHREIBUNG DER AUSFÜHRUNGSFORMEN

[0005] In der Computer-Vision beinhaltet das typische Rahmenwerk für das Verfolgen von Objekten ein Objekterkennungsmodul, ein Objektverfolgermodul und ein Objektidentifikationsmodul, die gemäß einem Objektverfolgungsmodell implementiert sind.

Das Objektverfolgungsmodell, das im Kontext von Langzeit-Objektverfolgung auch als Online-Objektverfolgungsmodell bezeichnet wird, ist eine Art Maschinelllernmodell, das einen Maschinellernalgorithmus und Trainingsdaten beinhaltet. Der Maschinellernalgorithmus erlernt aus den Trainingsdaten, ein Objektverfolgungsmodell zu erzeugen, um Objekte mit der Zeit zu verfolgen.

[0006] Die Objekterkennungs-, Objektverfolger- und Objektidentifikationsmodule, die gemäß dem Objektverfolgungsmodell implementiert sind, setzen einen Maschinelllernklassifizierer ein, um zwischen Objekten basierend auf Erscheinungsmerkmalen von Objekten zu unterscheiden, die mit der Zeit beobachtet werden. Der Maschinelllernklassifizierer ist ein mathematisches Modell eines Satzes von Daten, die die Erscheinungsmerkmale repräsentieren. Unter Verwendung von Convolutional Neural Networks ist der Maschinelllernklassifizierer in der Lage, verfolgte Objekte basierend auf den Daten zu identifizieren, die die Erscheinungsmerkmale repräsentieren.

[0007] Bezug nehmend auf **Fig. 1** initialisiert ein Objekterkennungsmodul **102**, um das Verfolgen zu beginnen, ein Objektverfolgermodul **108**, um ein bestimmtes Objekt zu verfolgen, das hierin als Zielobjekt **104** bezeichnet wird. Das Objekterkennungsmodul **102** stellt einen präzisen Hüllkörper für jedwedes vorgegebene Ziel bereit. Es kann mithilfe von Objektdetektoren nach Stand der Technik implementiert werden, wie z.B. FAST-R-CNN (Fast Region-Based Convolutional Network), Mask-R-CNN (Mask Region-Based Convolutional Network), SSD (Single Shot Detection Neural Network), YOLO (You-Only-Look-Once-Echtzeit-Objekterkennung) usw. Es kann auch über Benutzer-Interaktion implementiert werden.

[0008] Das Erkennungsmodul **102** initialisiert einen oder mehrere Verfolgungsparameter **120** des Objektverfolgermoduls **108** durch Erlernen eines oder mehrerer entsprechender Erscheinungsmerkmale des Zielobjektes **104** in einem Einzelbild, in dem das Zielobjekt erkannt wurde. Das Einzelbild bezeichnet ein Bild in einer Sequenz von Videoeinzelbildern oder einer Sequenz von Bildern, die in einem Videostream und dergleichen erfasst wurden.

[0009] Im Allgemeinen wird, sobald das Zielobjekt **104** mit einem Hüllkörper initialisiert worden ist, das Objektverfolgermodul **108** die Parameter **120** des Objektverfolgungsmodells **122** mit einer gegebenen Anmerkung initialisieren, die einem Erscheinungsmerkmal des Zielobjektes **104** entspricht. Die Parameter **120** des Objektverfolgermoduls hängen von der Art des verwendeten Objektverfolgungsmodells **122** ab. In diesem Fall entsprechen die Parameter **120** des Moduls den erlernten Erscheinungsmerkmalen des verfolgten Zielobjektes **104**. Die erlernten Erscheinungsmerkmale können solche Charakteristika

wie Farbe, Form und so fort beinhalten. Derartige Merkmale werden verwendet, um ein erkanntes Objekt von einem anderen mithilfe des Maschinelllernklassifizierers zu unterscheiden.

[0010] In nachfolgenden Einzelbildern sucht das Objektverfolgermodul **108** um die letzte bekannte Position des Zielobjektes **104** herum. Die letzte bekannte Position des Zielobjektes ist die Position nach der anfänglichen Erkennung oder die Position nach dem letzten Mal, dass das Zielobjekt in einem vorherigen Einzelbild erfolgreich verfolgt wurde. Das Objektverfolgermodul **108** versucht, das Zielobjekt an der letzten bekannten Position mithilfe des einen oder der mehreren erlernten Erscheinungsmerkmale des Zielobjektes zu finden. Beispielsweise kann das Zielobjekt **104** durch Maximieren einer Antwort des Objektverfolgungsmodells **122** und Verwenden der Antwort als Prognose der zukünftigen Position des Zielobjektes in einem nachfolgenden Einzelbild oder nachfolgenden Einzelbildern gefunden werden.

[0011] Das Objektidentifikationsmodul **106** ist konfiguriert zu prüfen, ob das Verfolgen des Zielobjektes **104** durch das Objektverfolgermodul **108** erfolgreich ist. In einigen Fällen kann das Zielobjekt in einem nachfolgenden Einzelbild nicht gefunden werden, wie z.B., wenn sich das Zielobjekt aus dem Einzelbild herausbewegt. In anderen Fällen kann das Zielobjekt **104** eine große Veränderung des Erscheinungsbildes aufgrund von Abdeckung, Verformung usw. erfahren, die bewirkt, dass das Objektverfolgermodul **108** die Spur des Zielobjektes verliert oder dass das Objektidentifikationsmodul **106** es nicht als das Zielobjekt identifizieren kann. So oder so wird, sobald das Zielobjekt **104** verloren ist, das Objektverfolgermodul **108** durch Löschen aller Verfolgungsmodellparameter **120** zurückgesetzt, die gesetzt wurden, als das Objektverfolgermodul **108** initialisiert wurde. Das Objekterkennungsmodul **102** reinitialisiert das Objektverfolgermodul **108**, um durch erneutes Erkennen des Zielobjektes **104** noch einmal von vorn zu beginnen.

[0012] Während der Objektverfolgung wird die Verfolgung dann fortgeführt, solange das Zielobjekt **104** gefunden wird. Anders ausgedrückt können, wenn das Zielobjekt **104** basierend auf den Parametereinstellungen des Objektverfolgermoduls **108** verfolgt und erfolgreich identifiziert wurde, mithilfe der Verfolgungsposition neue Muster, die das Zielobjekt **104** enthalten, aus den Einzelbildern erlangt werden, in denen sie gefunden werden. Die neuen Muster, die auch als Ausschnitte bezeichnet werden, die aus den Einzelbildern ausgeschnitten werden, werden verwendet, um das Objektverfolgungsmodell **122** zu aktualisieren **118**, um Verformung des Zielobjektes Rechnung zu tragen. Beispielsweise werden, wenn sich ein Erscheinungsbild des Zielobjektes **104** mit der Zeit ändert, die neuen Muster/Ausschnitte, die

das Zielobjekt erfassen, verwendet, um die Parameter **120** des Objektverfolgermoduls **108** zu aktualisieren, um die erlernten Erscheinungsmerkmale des Zielobjektes einschließlich der Verformung des Zielobjektes widerzuspiegeln.

[0013] Eines der herausforderndsten Probleme beim Implementieren von Objektverfolgung unter Verwendung des oben beschriebenen Rahmenwerkes (Objekterkennungsmodul **102**, Objektidentifikationsmodul **106** und Objektverfolgermodul **108** gemäß einem Objektverfolgungsmodell **122** und zugeordneter Parameter **120**) ist die Wanderung in der Verfolgungsposition des Zielobjektes **104** mit der Zeit. Wanderung ist typischerweise bedingt durch Fehlerfortpflanzung aus dem Aktualisieren des Objektverfolgungsmodells **122**, implementiert durch das Objektverfolgermodul **108** mit den neuen Mustern/Ausschnitten, die das Zielobjekt **104** erfassen. Tritt Wanderung auf, ist das Verfolgungsergebnis, das durch das Objektverfolgermodul **108** erlangt wird, gegenüber der wahren Position des Zielobjektes **104** versetzt. Als Folge davon enthalten die neuen Muster/Ausschnitte, die unter Verwendung der „gewanderten“ Zielposition statt der wahren Position ausgeschnitten werden, unrichtige Informationen über das Zielobjekt **104**.

[0014] Neben anderen Fehlern erhöht Wanderung die Wahrscheinlichkeit, dass das Objektverfolgermodul **108** einiges des Hintergrundes in den neuen Mustern/Ausschnitten, die das Zielobjekt **104** erfassen, als Vordergrund interpretiert und umgekehrt. Mit der Aggregation weiterer Muster/Ausschnitte lernt das Objektverfolgermodul **108** unrichtiges Wissen über das Zielobjekt **104**. Als Konsequenz daraus sucht das Objektverfolgermodul **108** in nachfolgenden Einzelbildern am falschen Ort nach dem Zielobjekt **104**. Anders ausgedrückt, sucht das Objektverfolgermodul **108** an der „gewanderten“ Position und entfernt von der wahren Position nach dem Zielobjekt **104**. Das Suchen am falschen Ort verschlimmert die Wanderung weiter und resultiert schließlich in Verfolgungsversagen. Aus diesem Grunde ist Objektverfolgung in aktuellen Computer-Vision-Anwendungen nur über kurze Zeiträume genau, typischerweise weniger als eine Minute lang.

[0015] Zum Beispiel unterstützt die Visual-Object-Tracking- (VOT-) - Datensatzbibliothek zum Ermitteln von Benchmarks in der Objektverfolgung (<http://www.votchallenge.net>) Verfolgung mit einer Dauer von durchschnittlich nur 7 bis 30 Sekunden. Derartige zeitliche Einschränkungen beeinträchtigen die Fähigkeit, Objektverfolgung in realen Situationen zu verwenden, die anfällig für Fehler sind aufgrund von beispielsweise Objektverdeckung (z.B. durch andere Objekte im Einzelbild), der Bewegung des Objektes aus dem Einzelbild heraus und/oder dem Vorhandensein eines ähnlichen Erscheinungsmerkmals

(z.B. einer ähnlichen Farbe) des Objektes im Hintergrund, das bewirkt, dass das Objekt schlecht zu „sehen“ ist, und dergleichen. Um ein Objektverfolgungsmodell **122** an praktische Anwendungen in der Realität anzupassen, ist das Objektverfolgermodul **108** gezwungen, wenn die Verfolgung versagt, oder in regelmäßigen Zeitintervallen Wanderung durch Zurücksetzen und Neustarten Rechnung zu tragen.

[0016] Zurücksetzen und Neustarten des Objektverfolgermoduls **108** verwirft jedoch möglicherweise wertvolle Muster, die vor dem Verfolgungsversagen erlangt wurden. Verwerfen von Mustern verursacht, dass das Objektverfolgermodul **108** bereits erlernte Informationen vergisst, und verschlechtert die Fähigkeit des Objektverfolgungsmodells **122**, zwischen möglichen Kandidatenobjekten für Langzeit-Objektverfolgung zu unterscheiden.

[0017] Um das Problem der Wanderung in der Verfolgungsposition bei Langzeit-Objektverfolgung anzugehen, beinhalten Ausführungsformen kontinuierlichen Lernens für Langzeitverfolgung ein Rahmenwerk für kontinuierliches Lernen **100**. Statt Muster, die erlernte Erscheinungsmerkmale des Zielobjektes enthalten, beim Zurücksetzen und Neustarten des Objektverfolgermoduls **108** zu verwerfen, verbessert in einer Ausführungsform das Rahmenwerk für kontinuierliches Lernen **100** das Verfolgen durch kontinuierliches Akkumulieren von Mustern in großem Maßstab und Speichern der akkumulierten Muster in einer Datenbank der Muster kontinuierlichen Lernens **112**.

[0018] In einer Ausführungsform aktualisiert ein Modul für kontinuierliches Lernen **110** mithilfe der akkumulierten Muster, die in der Datenbank der Muster kontinuierlichen Lernens **112** gespeichert sind, kontinuierlich das aktuelle Objektverfolgungsmodell **122** basierend auf den erlernten Erscheinungsmerkmalen, die in den akkumulierten Mustern enthalten sind. Durch Verwenden eines kontinuierlich aktualisierten Objektverfolgungsmodells **122** vermeidet das Modul für kontinuierliches Lernen **110** das Einführen von Mustern schlechter Qualität, die ungenaue Erscheinungsmerkmalen enthalten, wodurch Fehlerfortpflanzung begrenzt wird. In einer Ausführungsform beinhaltet ein aktuelles Objektverfolgungsmodell **122**, das fähig ist, kontinuierlich aktualisiert zu werden, ein diskriminatives Korrelationsfilter (Discriminative Correlation Filter, DCF), das faktorisierte Faltungsoperatoren verwendet, um die Anzahl der Parameter im Objektverfolgungsmodell **122** zu verringern.

[0019] Fig. 2 ist eine schematische Blockdiagrammdarstellung einer Architektur **200** zum Implementieren kontinuierlichen Lernens zur Langzeit-Objektverfolgung gemäß verschiedenen hierin beschriebenen Beispielen. In einer Ausführungsform wird bei Prozess **202** ein Objekterkennungsmodul **102** mit dem

Zielobjekt **104** initialisiert. Beispielsweise kann das Zielobjekt **104** mit einer manuellen Interaktion ausgewählt werden oder kann das Objekterkennungsmodul **102** das Zielobjekt **104** automatisch erkennen einschließlich Wiedererkennen des Zielobjektes, wobei Letzteres typischerweise auf vorherigen manuellen und/oder Trainingserkennungen des Zielobjektes **104** basiert.

[0020] Bei Prozess **204** wird das Objektverfolgermodul **108** mit dem erkannten (oder wiedererkannten) Zielobjekt **104** und jedweden etikettierten Anmerkungsparametern initialisiert, die während der Erkennung erlangt werden. Die etikettierten Anmerkungsparameter entsprechen den erlernten Erscheinungsmerkmalen des Zielobjektes **104**. Bei Prozess **206** wird das Objektverfolgermodul **108** mit der Datenbank der Muster kontinuierlichen Lernens **112** mit Mustern aktualisiert, die während vorherigen erfolgreichen Verfolgens akkumuliert wurden, einschließlich jedweder etikettierten Anmerkungen des Zielobjektes, die in den akkumulierten Mustern gefunden werden.

[0021] In einer Ausführungsform verfolgt bei Prozess **208** das Objektverfolgermodul **108** unter Verwendung des Zielobjektes **104** mögliche Kandidatenobjekte in nachfolgenden Einzelbildern der Videobilder, wie z.B. während des Trainings und/oder während nachfolgender Langzeit-Objektverfolgung. Das Zielobjekt **104** kann das anfängliche Zielobjekt aus Prozess **202** oder ein wiedererkanntes Zielobjekt aus Prozess **212** sein. Das/die Bild(er) **210**, in denen das Zielobjekt **104** verfolgt wird, beinhalten ein oder mehrere Videobilder, die auch als Einzelbilder bezeichnet werden, aus irgendeiner oder mehreren Ansichten (Mehrfachansichten), die durch mehrere Bildfassungsgenäte erzeugt werden. Das/die Bild(er) **210** können auch Mehrfachinstanzen eines einzigen Bildes beinhalten.

[0022] In einer Ausführungsform untersucht bei Prozess **214** das Objektidentifikationsmodul **106** verfolgte Kandidatenobjekte. Beispielsweise kann das Objektidentifikationsmodul **106** die Merkmale eines verfolgten Kandidatenobjektes mit dem Zielobjekt **104** sowie den verfügbaren Ansichten des Zielobjektes **104** vergleichen, die in den akkumulierten Mustern identifiziert worden sind, die in der Datenbank der Muster kontinuierlichen Lernens **112** gespeichert sind, um Mehrfachansichtsähnlichkeit zu untersuchen.

[0023] In einer Ausführungsform beinhaltet das Objektidentifikationsmodul **106** ein Wiederidentifikationsnetz, um zu unterscheiden, ob ein verfolgter Bereich das Zielobjekt **104** enthält. Beispielsweise extrahiert das Wiederidentifikationsnetz Erscheinungsmerkmale durch Weiterleiten beobachteter Merkmale im verfolgten Bereich an ein Deep-Neural-Net-

work. Das Objektidentifikationsmodul **106** berechnet den Abstand zwischen den extrahierten Erscheinungsmerkmalen, die im verfolgten Bereich vorhanden sind, und den Erscheinungsmerkmalen, die im Zielobjekt **104** vorhanden sind. Das Objektidentifikationsmodul **106** berechnet ferner den Abstand zwischen den extrahierten Erscheinungsmerkmalen, die im verfolgten Bereich vorhanden sind, und den Erscheinungsmerkmalen, die in den akkumulierten Mustern vorhanden sind, die in der Datenbank der Muster kontinuierlichen Lernens **112** für das Zielobjekt **104** gespeichert sind.

[0024] In Entscheidungsprozess **216** ist, wenn der durchschnittliche Abstand zwischen den extrahierten Erscheinungsmerkmalen, die im verfolgten Bereich vorhanden sind, und den Erscheinungsmerkmalen, die in Zielobjekt **104** vorhanden sind, und akkumulierten Mustern kleiner als eine Identitätsschwelle ist, das Verfolgungsergebnis erfolgreich, was die Erfolgswahrscheinlichkeit beim Fortfahren erhöht, das Zielobjekt **104** ohne Zurücksetzen des Objektverfolgermoduls **108** zu verfolgen, einschließlich ohne das Zielobjekt **104** wiedererkennen zu müssen. Das Verfolgungsergebnis ist eine Verfolgungsantwortkarte des Kandidatenobjektes im verfolgten Bereich. Auf erfolgreiches Verfolgen hin kann bei Prozess **218** das Modul für kontinuierliches Lernen **110** damit fortfahren, ein Muster aus einem Einzelbild auszuscheiden, in dem ein erfolgreich verfolgtes Kandidatenobjekt erscheint, und alle derartig ausgeschnittenen Muster in einem Datensatz des Musters kontinuierlichen Lernens für das Zielobjekt **104** zur letztendlichen Speicherung in der Datenbank der Muster kontinuierlichen Lernens **110** zu erfassen.

[0025] Unter gewissen Umständen wird Objektverfolgung als erfolglos angesehen. Beispielsweise geht, wenn der Durchschnitt der Abstände zwischen den extrahierten Erscheinungsmerkmalen, die im verfolgten Bereich vorhanden sind, und den Erscheinungsmerkmalen, die in Zielobjekt **104** vorhanden sind, und akkumulierten Mustern größer als die Identitätsschwelle ist, das Objektverfolgermodul **108** verloren. Das Zielobjekt **104** muss durch das Objekterkennungsmodul **102** wiedererkannt werden, bevor weitere Objektverfolgung stattfinden kann. Somit fährt bei Prozess **212** das Objekterkennungsmodul **102** damit fort, das Zielobjekt **104** im aktuellen Einzelbild wiederzuerkennen.

[0026] Auch wenn das Verfolgen erfolglos ist, kann, weil das Objektverfolgermodul **108** mit den ausgeschnittenen Mustern im Datensatz des Musters kontinuierlichen Lernens aus einem oder mehreren vorherigen erfolgreichen Verfolgungsergebnissen aktualisiert worden ist, einschließlich jedweder etikettierten Anmerkungsparameter, die die ausgeschnittenen Muster begleiten, das Objektverfolgermodul **108** das wiedererkannte Zielobjekt **104** verwenden, um Lang-

zeitverfolgung fortzusetzen, ohne die wertvollen Informationen in den akkumulierten Mustern zu verlieren, die in der Datenbank der Muster kontinuierlichen Lernens **112** gespeichert sind.

[0027] Fig. 3 ist eine schematische Blockdiagrammdarstellung einer Übersicht **300** eines Modul für kontinuierliches Lernen **110** zum Implementieren kontinuierlichen Lernens zur Langzeit-Objektverfolgung gemäß vershierten hierin beschriebenen Beispielen. In einer Ausführungsform beinhaltet das Modul für kontinuierliches Lernen **110** einen Musterpräprozessor **302** und einen Musterausschneider **304**.

[0028] In einer Ausführungsform ist der Musterpräprozessor **302** konfiguriert, einen Ort eines Kandidatenobjektes, das im Verfolgungsergebnis des Objektverfolgermoduls **108** gefunden wurde, mithilfe beispielsweise eines Verfeinerungsnetzes und/oder von Clustering der akkumulierten Muster aus der Datenbank der Muster kontinuierlichen Lernens **112** zu verfeinern. Sobald der Ort eines Kandidatenobjektes verfeinert worden ist, schneidet ein Musterausschneiderprozessor **304** einen Ausschnitt aus dem Gesamtbild aus, um das Muster zu bilden, und fügt das Muster zum Merken aller Veränderungen der Verformungen des Zielobjektes **104**, die in den ausgeschnittenen Mustern repräsentiert sind, in eine Datensatz des Musters kontinuierlichen Lernens ein.

[0029] Die oben beschriebene Architektur kontinuierlichen Lernens für Langzeit-Objektverfolgung, die in Fig. 2-3 dargestellt ist, wird in dem Beispiel demonstriert, das in Fig. 4 dargestellt ist. Eine Reihe von Eingangseinzelbildern EINZELBILD 0, EINZELBILD 1, EINZELBILD 8, EINZELBILD 11, EINZELBILD 17 und EINZELBILD 36 enthält Videobilder eines Boxkampfes zwischen zwei Boxern und eines Ringrichters, der den Boxkampf beaufsichtigt. Ein Objekterkennungsmodul **102**, Objektidentifikationsmodul **106** und Objektverfolgermodul **108** erzeugen eine anfängliche Verfolgungsantwortkarte **402**, die die Boxer und den Ringrichter identifiziert. Nur als Beispiel ist das anfängliche Zielobjekt **404** der Ringrichter, wie in EINZELBILD 0 initialisiert und wie in den Verfolgungsantwortkarten **405** verfolgt, die durch das Objektverfolgermodul **108** aus nachfolgenden Einzelbildern EINZELBILD 1, EINZELBILD 8, EINZELBILD 11, EINZELBILD 17 und EINZELBILD 36 erzeugt worden sind.

[0030] In einer Ausführungsform wird ein Satz von fünf Verfolgungsergebnisausschnitten **408** aus den Einzelbildern ausgeschnitten, die Kandidatenobjekte erfassen, die den verfolgten Ringrichter in verschiedenen Posen und mit verglichen mit dem Erscheinungsbild des verfolgten Ringrichters im anfänglichen Zielobjekt **404** sich verändernden Mengen von Zielverformungen repräsentieren.

[0031] In einer Ausführungsform wird ein Teilsatz von vier Verfolgungsergebnisausschnitten **408** als Muster kontinuierlichen Lernens **410** in einem Datensatz der Muster kontinuierlichen Lernens erfasst. Ein fünfter Verfolgungsergebnisausschnitt **408** (der in der Mitte des Satzes von fünf Verfolgungsergebnisausschnitten erscheint) wird nicht in den Mustern kontinuierlichen Lernens **410** erfasst, da er ein zu kleines Segment des Zielobjektes **104** (die Schulter des verfolgten Ringrichters) enthält, um von Wert zu sein. Die erfassten Muster kontinuierlichen Lernens **410** zeigen das Erscheinungsbild des Zielobjektes **404**, des verfolgten Ringrichters, unter unterschiedlichen Umständen. Beispielsweise sind, wenn der verfolgte Ringrichter durch einen der Boxer verdeckt ist, gewisse Segmente des Ringrichters dennoch sichtbar und können für zukünftige Verfolgung wertvoll sein. In einer Ausführungsform werden die erfassten Muster kontinuierlichen Lernens **410** in der Datenbank der Muster kontinuierlichen Lernens **112** zur nachfolgenden Verwendung während des Trainings und/oder während Langzeitverfolgung gespeichert.

[0032] In einer Ausführungsform werden die Muster kontinuierlichen Lernens **410** im Objektidentifikationsmodul **106** verwendet, um zu ermitteln, ob das Kandidatenobjekt in einem bestimmten Verfolgungsergebnis tatsächlich das Zielobjekt **104/404** ist. In einer Ausführungsform implementiert das Objektidentifikationsmodul **106** durch Berechnen des durchschnittlichen Abstands zwischen den Erscheinungsmerkmalen des Zielobjektes **104/404** und jedem der Erscheinungsmerkmale, die aus den Mustern kontinuierlichen Lernens **410** extrahiert wurden, wie ausführlicher unter Bezug auf Fig. 2-3 beschrieben, einen Algorithmus für Mehrfachansichtsvergleiche, der robust gegenüber Störungen ist.

[0033] Im dargestellten Beispiel in Fig. 4 sind die vier optimalen ausgeschnittenen Ausschnitte, die das Kandidatenobjekt erfassen und als Muster kontinuierlichen Lernens **410** erfasst sind, jene, die die besten Ansichten des verfolgten Ringrichters enthalten. Die besten Ansichten in den Mustern kontinuierlichen Lernens **410** sind jene, in denen die durchschnittlichen Abstände zwischen Erscheinungsmerkmalen des Kandidatenobjektes verglichen mit den Erscheinungsmerkmalen des Zielobjektes **104** hinreichend klein sind, um das Kandidatenobjekt als Zielobjekt **104** zu identifizieren. Im dargestellten Beispiel nach Fig. 4 ist das Zielobjekt **104**, das zum Vergleich verwendet wird, der verfolgte Ringrichter/ das anfängliche verfolgte Objekt **404**. In einer Ausführungsform werden die Abstände zwischen den Erscheinungsmerkmalen unter Verwendung eines Identitätsschwellenabstandes berechnet. Der Identitätsschwellenabstand kann jedwedes von vorgegeben und/oder dynamisch konfiguriert sein, basierend beispielsweise auf einer Größe des Zielobjektes **104**,

das verfolgt wird, relativ zu anderen Objekten in einem Eingangseinzelbild.

[0034] Im dargestellten Beispiel in **Fig. 4** kann der Musterpräprozessor **302** nach **Fig. 3** mit einem Segmentierungsmodul konfiguriert sein, um den Ort des verfolgten Ringrichters zu verfeinern. In einer Ausführungsform wird lediglich als Beispiel und nicht als Einschränkung das Segmentierungsmodul durch den Musterpräprozessor **302** verwendet, um die Muster kontinuierlichen Lernens **410** mit einem Segmentierungskontur-Erscheinungsmerkmal zu etikettieren, das die Konturen verschiedener Segmente umreißt, die das anfängliche Zielobjekt **404**, das verfolgt wird, und die Kandidatenobjekte umfassen, die erfolgreich zum anfänglichen Zielobjekt **404** führen. Natürlich können andere Präprozessingverfahren verwendet werden, um die Qualität der Muster kontinuierlichen Lernens **410** zu verbessern.

[0035] Die Muster kontinuierlichen Lernens **410**, die im Datensatz des Musters kontinuierlichen Lernens akkumuliert und in der Datenbank der Muster kontinuierlichen Lernens **112** gespeichert sind, sind wertvoll, weil sie die Etikettierungsinformationen (auch als Anmerkungen bezeichnet) für das Zielobjekt **104/404** und Kandidatenobjekte ohne manuelle Intervention transportieren. Die Muster kontinuierlichen Lernens **410** werden in den verschiedenen Ausführungsformen kontinuierlichen Lernens für Langzeit-Objektverfolgung verwendet, um Erkennungs- und Identifikationsgenauigkeit zu verbessern.

[0036] **Fig. 5A-5B** sind schematische Blockdiagramme, die kontrastierende Schemata für das Trainieren für Langzeit-Objektverfolgung mit kontinuierlichem Lernen und ohne dieses gemäß verschiedenen hierin beschriebenen Beispielen darstellen. Beispielsweise stellt **Fig. 5A** einen Standard-Verfolgungsalgorithmus dar, der einen Trainingsprozess mit geschlossener Schleife **502** benutzt, in dem ein Muster extrahiert und als Trainingsmuster verwendet wird. Während des Ablaufs des Trainingsprozesses mit geschlossener Schleife **502** wird ein diskriminativer Klassifizierer **504** für ein anfängliches Zielobjekt auf ein Eingangseinzelbild **506** angewandt, um eine maximale Antwort als Trainingsmuster **508** zu extrahieren. Während des Trainings **510** wird wiederum das Trainingsmuster **508** verwendet, um den diskriminativen Klassifizierer **504** zu aktualisieren.

[0037] Im Gegensatz dazu werden in einem Trainingsprozess mit offener Schleife **512** mit kontinuierlichem Lernen mehrere Trainingsmuster verwendet, um Genauigkeit und Identifikation bei Langzeit-Objektverfolgung zu verbessern. Beispielsweise wird, um die geschlossene Schleife aufzubrechen, ein Objektwiederidentifikationsmodul (z.B. das Objektidentifikationsmodul **106**) eingeführt, um dem Objektverfolgermodul **108** zu helfen, in nachfolgenden Einzelbil-

dern einen genaueren Ort des verfolgten Zielobjektes **104** zu finden. In einer Ausführungsform kann während Langzeit-Objektverfolgung das Objektwiederidentifikationsmodul **106** periodisch aufgerufen werden, wie z.B. alle N Einzelbilder, um die Genauigkeit des Objektverfolgungsmodells **122** und der Parameter **120** zu erhöhen, die verwendet werden, um das Objektverfolgermodul **108**, Objektidentifikationsmodul **106** und Objekterkennungsmodul **102** zu implementieren.

[0038] Beispielsweise werden während Trainings mit offener Schleife mit kontinuierlichem Lernen **512** das aktuelle verfolgte Objekt **514** (das Kandidatenobjekt) und das aktuelle erkannte Objekt **516** (das Zielobjekt) beide im Wiederidentifikationsmodul **106** verwendet, um genaue Trainingsmuster während Training **510** wie auch während Langzeit-Objektverfolgung zu finden. Die genauen Trainingsmuster werden in der Datenbank der Muster kontinuierlichen Lernens **112** zur späteren Verwendung während Anwendung des diskriminatorischen Klassifizierers **504** auf nachfolgende Einzelbilder einer Sequenz von Videoeinzelbildern gespeichert.

[0039] In einem Beispielszenario könnte sich ein Zielobjekt **104** während Langzeitverfolgung zu viel verformen, sodass Wiederidentifikation mit der ersten Anmerkung, mit der das anfängliche Zielobjekt **104** etikettiert ist, nicht möglich sein könnte. Dieses Szenario wird im Trainingsbeispiel **600** demonstriert, das in **Fig. 6A-6E** dargestellt ist. Ein Objektverfolgermodul **108** versucht, eine weiß gekleidete Straßenkünstlerin zu verfolgen, wie in **Fig. 6A-6E** in entsprechenden Einzelbildern EINZELBILD 2, EINZELBILD 3, EINZELBILD 8, EINZELBILD 9 und EINZELBILD 10 dargestellt.

[0040] Lediglich als Beispiel und nicht als Einschränkung sind in **Fig. 6A-6E** die Ergebnisse der Verfolgung mit Training mit geschlossener Schleife **502** den Ergebnissen der Verfolgung mit Training mit offener Schleife mit kontinuierlichem Lernen **512** gegenübergestellt. Die Ergebnisse der geschlossenen Schleife **602a** beginnen von den Ergebnissen der offenen Schleife **602b** ab EINZELBILD 8 in **Fig. 6C** abzuweichen, wie durch die durchgezogene Grenzlinie gekennzeichnet, die einen Teil des Einzelbildes abgrenzt, der das verfolgte Objekt enthält, in diesem Beispiel die Straßenkünstlerin. Die Ergebnisse der geschlossenen Schleife **602a** wandern von der Straßenkünstlerin nach rechts, wobei der diskriminatorische Klassifizierer **504** möglicherweise die weiße Farbe der Tür im Hintergrund mit der weißen Kleidung der Straßenkünstlerin verwechselt, wenn das Trainingsmuster ohne die Vorteile des kontinuierlichen Lernens verwendet wird.

[0041] Im Gegensatz dazu behalten die Ergebnisse der offenen Schleife **602b** die Verfolgung der Stra-

ßenkünstlerin bei, wie durch die gepunktete Grenzlinie gekennzeichnet, die einen Teil des Einzelbildes abgrenzt, der das verfolgte Objekt enthält, also die Straßenkünstlerin. Beispielsweise ist in EINZELBILD 8 der Fig. 6C und EINZELBILD 9 der Fig. 6D die Straßenkünstlerin nahezu perfekt innerhalb der gepunkteten Grenzlinie zentriert. Ab EINZELBILD 10 der Fig. 6E wandern die Ergebnisse der geschlossenen Schleife 602a weiter von der Straßenkünstlerin nach rechts, wohingegen die Ergebnisse der offenen Schleife 602b weiterhin nahezu perfekt innerhalb der gepunkteten Grenzlinie zentriert sind. Jedes der Ergebnisse der offenen Schleife 602b wird als Trainingsmuster kontinuierlichen Lernens gespeichert, die aus EINZELBILD 0, EINZELBILD 3, EINZELBILD 8, EINZELBILD 9 und EINZELBILD 10 in der jeweiligen Fig. 6A-6E ausgeschnitten worden sind. Die Ergebnisse der offenen Schleife 602b als Verfolgungsfortschritt bis EINZELBILD 10 in Fig. 6E zeigen an, dass der diskriminatorische Klassifizierer 504 von den Trainingsmustern kontinuierlichen Lernens profitiert, um die Genauigkeit der Erkennung und Identifikation während Langzeit-Objektverfolgung zu verbessern.

[0042] Das oben beschriebene Szenario der Fig. 6A-6E ist weiter in Trainingsbeispiel 700 dargestellt, das in Fig. 7A-7D und jeweils in EINZELBILD 22, EINZELBILD 37, EINZELBILD 54 und EINZELBILD 67 dargestellt ist. Die Standard-Verfolgungsalgorithmus-Ergebnisse der geschlossenen Schleife 702a und Ergebnisse der offenen Schleife mit kontinuierlichem Lernen 702b werden ferner Ergebnissen einer Verfolgung mit Rücksetz-Algorithmus 702c gegenübergestellt. Wie dargestellt, wandern die Standard-Verfolgungsalgorithmus-Ergebnisse der geschlossenen Schleife 702a weiterhin nach rechts, und die Verfolgung der Straßenkünstlerin ist in Fig. 7A-7D verloren. Die Verfolge- und Rücksetz-Algorithmus-Ergebnisse 702c verfolgen weiterhin die Straßenkünstlerin in Fig. 7A-7D, aber ab EINZELBILD 37 in Fig. 7B beginnt der Verfolge- und Rücksetz-Algorithmus 702c, einen Teil des rechten Armes der Straßenkünstlerin zu verlieren. Ab EINZELBILD 67 in Fig. 7D beginnt der Verfolge- und Rücksetz-Algorithmus 702c, ebenso einen Teil des Rückens der Straßenkünstlerin zu verlieren. Im Gegensatz dazu behalten die Ergebnisse der offenen Schleife mit kontinuierlichem Lernen 702b weiterhin einen genauen Ort der Straßenkünstlerin bei und vermeiden das Abschneiden wesentlicher Teile der Straßenkünstlerin.

[0043] Fig. 8 ist eine Darstellung eines Systems, in dem kontinuierliches Lernen für Langzeit-Objektverfolgung gemäß einer Ausführungsform implementiert sein kann. In dieser Darstellung sind gewisse standardmäßige und wohlbekannte Komponenten nicht gezeigt, die für die vorliegende Beschreibung nicht relevant sind. Elemente, die als getrennte Elementen

te gezeigt sind, können kombiniert sein, einschließlich beispielsweise einem SoC (System on Chip), das mehrere Elemente auf einem einzigen Chip kombiniert.

[0044] In einigen Ausführungsformen kann ein Computing-System wie z.B. Plattform 800 ein Verarbeitungsmittel wie z.B. einen oder mehrere Prozessoren 810 beinhalten, die mit einem bzw. einer oder mehreren Bussen oder Verbindungen gekoppelt sind, die im Allgemeinen als Bus 805 gezeigt sind. Die Prozessoren 810 können einen oder mehrere physikalische Prozessoren und einen oder mehrere logische Prozessoren umfassen. In einigen Ausführungsformen können die Prozessoren einen oder mehrere Mehrzweckprozessoren oder Spezialprozessoren beinhalten.

[0045] Der Bus 805 ist ein Kommunikationsmittel zur Übertragung von Daten. Lediglich als Beispiel und nicht als Einschränkung ist der Bus 805 als einzelner Bus dargestellt, kann aber mehrere unterschiedliche Verbindungen oder Busse repräsentieren, und die Komponentenverbindungen zu derartigen Verbindungen können variieren. Der in Fig. 8 gezeigte Bus 805 ist eine Abstraktion, die eine(n) oder mehrere beliebige separate physikalische Busse, Punkt-zu-Punkt-Verbindungen oder beides repräsentiert, verbunden durch zweckentsprechende Bridges, Adapter oder Steuerungen.

[0046] In einigen Ausführungsformen umfasst das Computing-System 800 ferner einen Direktzugriffsspeicher (Random Access Memory, RAM) oder ein anderes dynamisches Speichergerät oder -element als Hauptspeicher 815 und Speichersteuerung 816 zum Speichern von Informationen und Befehlen, die durch die Prozessoren 810 auszuführen sind. Hauptspeicher 815 kann, ohne jedoch darauf beschränkt zu sein, dynamischen Schreib-Lese-Speicher (Dynamic Random Access Memory, DRAM) beinhalten. In einigen Ausführungsformen beinhaltet der RAM oder das andere dynamische Speichergerät oder -element einen Verfolgungsmodell-Schaltungsblock 818, der Logik kontinuierlichen Lernens 817 implementiert, um der Verfolgungsmodelllogik 818 Trainingsmuster zum Ausführen von Langzeitverfolgung gemäß den beschriebenen Ausführungsformen bereitzustellen.

[0047] Das Computing-System 800 kann auch einen nichtflüchtigen Speicher 820; ein Speichergerät wie z.B. ein Halbleiterlaufwerk (Solid-State Drive, SSD) 830 und einen Nur-Lese-Speicher (Read Only Memory, ROM) 835 oder ein anderes statisches Speichergerät zum Speichern statischer Informationen und Befehle für die Prozessoren 810 umfassen.

[0048] In einigen Ausführungsformen beinhaltet das Computing-System 800 einen oder mehrere Sender oder Empfänger 840, die mit dem Bus 805 gekoppelt

sind. In einigen Ausführungsformen kann das Computing-System **800** eine oder mehrere Antennen **844**, wie z.B. Dipol- oder Monopolantennen, für die Sendung und den Empfang von Daten per Drahtloskommunikation mithilfe eines Drahtlossenders, -empfängers oder beider und einen oder mehrere Anschlüsse **842** für die Sendung und den Empfang von Daten per leitungsgebundener Kommunikation beinhalten. Zur Drahtloskommunikation zählen ohne jedoch darauf beschränkt zu sein, Wi-Fi, Bluetooth™, Nahfeldkommunikation und andere Drahtloskommunikations-Standards.

[0049] In einigen Ausführungsformen beinhaltet Computing-System **800** ein oder mehrere Eingabegeräte **850** für die Eingabe von Daten, einschließlich Tasten oder Schaltflächen mit fester oder variabler Funktion, eines Joysticks, einer Maus oder eines anderen Zeigegerätes, einer Tastatur, eines Sprachbefehlssystems oder eines Gestenerkennungssystems. In einer Ausführungsform beinhaltet das Computing-System **800** eine(n) oder mehrere Tiefenkameras/-scanner **852**, um Daten zu erfassen, die zur Bildverarbeitung verwendet werden, einschließlich 3D-Daten. Die Tiefenkameras/-scanner können RGB-D-Sensoren (Rot-, Grün-, Blau-, Tiefensensoren) und Lidar-Fernabtaster (Light Detection and Ranging, Licht-Erkennung und -Entfernungsbestimmung) beinhalten.

[0050] In einigen Ausführungsformen beinhaltet Computing-System **800** eine Ausgabeanzeige **855**, wobei die Ausgabeanzeige **855** eine Flüssigkristallanzeige (Liquid Crystal Display, LCD) oder jedwede andere Anzeigetechnologie beinhalten kann, zum Anzeigen von Informationen oder Inhalten für einen Benutzer. In einigen Umgebungen kann die Ausgabeanzeige **855** einen Berührbildschirm beinhalten, der auch als mindestens ein Teil eines Eingabegerätes **850** genutzt wird. Ausgabeanzeige **855** kann ferner Audioausgabe einschließlich eines/einer oder mehrerer Lautsprecher, Audioausgangsbuchsen oder anderes Audio und andere Ausgabe für den Benutzer beinhalten.

[0051] Auch kann das Computing-System **800** eine Batterie oder andere Stromquelle **860** umfassen, die eine Solarzelle, eine Brennstoffzelle, einen geladenen Kondensator, eine induktive Nahfeldkopplung oder ein anderes System oder Gerät zum Bereitstellen oder Erzeugen von Strom im Computing-System **800** beinhalten kann. Der durch die Stromquelle **860** bereitgestellte Strom kann erforderlichenfalls an Elemente des Computing-Systems **800** verteilt werden.

[0052] Aus dieser Beschreibung wird offensichtlich, dass Aspekte der beschriebenen Ausführungsformen mindestens teilweise in Software implementiert sein könnten. Das heißt, dass die hierin beschriebenen Techniken und Verfahren in einem Datenverarbei-

tungssystem in Reaktion darauf ausgeführt werden könnten, dass sein Prozessor eine Abfolge von Befehlen ausführt, die in einem dinghaften, nichtflüchtigen Speicher wie z.B. dem Speicher **815** oder dem nichtflüchtigen Speicher **820** oder einer Kombination derartiger Speicher enthalten sind und wobei jeder dieser Speicher eine Form eines maschinenlesbaren, dinghaften Speichermediums ist.

[0053] Um die verschiedenen Ausführungsformen zu implementieren, könnten festverdrahtete Schaltungen in Kombination mit Softwarebefehlen verwendet werden. Beispielsweise können Aspekte der beschriebenen Ausführungsformen als Software implementiert werden, die in einem persistenten Speichergerät installiert und gespeichert ist und die durch einen Prozessor (nicht gezeigt) in einen Speicher geladen und ausgeführt werden kann, um die Prozesse oder Operationen auszuführen, die in dieser ganzen Anmeldung beschrieben sind. Alternativ können die beschriebenen Ausführungsformen mindestens zum Teil als ausführbarer Code implementiert sein, der in dedizierte Hardware programmiert oder eingebettet ist, wie z.B. eine integrierte Schaltung (z.B. einen anwendungsspezifischen IC oder ASIC), einen Digitalsignalprozessor (DSP), eine feldprogrammierbare Gatteranordnung (FPGA) oder Steuerung, auf die über einen entsprechenden Treiber und/oder ein Betriebssystem von einer Anwendung zugegriffen werden kann. Außerdem können die beschriebenen Ausführungsformen mindestens zum Teil als spezifische Hardwarelogik in einem Prozessor oder Prozessorkern als Teil eines Befehlssatzes implementiert sein, worauf durch eine Softwarekomponente über einen oder mehrere spezifische Befehle zugegriffen werden kann.

[0054] Somit sind die Techniken nicht auf irgendeine spezifische Kombination von Hardwareschaltung und Software oder auf irgendeine bestimmte Quelle für die Befehle beschränkt, die durch das Datenverarbeitungssystem ausgeführt werden.

[0055] Alle oder ein Teil der beschriebenen Ausführungsformen können mit Logikschaltungen implementiert sein, wie z.B. die oben beschriebenen ASIC-, DSP- oder FPGA-Schaltungen, einschließlich einer dedizierten Logikschaltung, einer Steuerung oder eines Mikrocontrollers oder einer anderen Form von Verarbeitungskern, der Programmcodebefehle ausführt. Somit könnten Prozesse, die durch die obige Diskussion gelehrt werden, mit Programmcode wie z.B. maschinenausführbaren Befehlen durchgeführt werden, die eine Maschine, die diese Befehle ausführt, veranlassen, gewisse Funktionen auszuüben. In diesem Kontext ist eine „Maschine“ typischerweise eine Maschine, die Befehle in einer Zwischenform (oder „abstrakte“ Befehle) in prozessor-spezifische Befehle umwandelt (z.B. eine abstrakte Laufzeitumgebung wie z.B. eine „virtuelle Maschi-

ne“ (z.B. eine virtuelle Java-Maschine), ein Interpreter, eine Common Language Runtime, eine virtuelle Hochsprachen-Maschine usw.), und/oder elektronische Schaltungen, die auf einem Halbleiterchip angeordnet (z.B. „Logikschaltungen“, die mit Transistoren implementiert sind) und ausgelegt sind, Befehle auszuführen, wie z.B. ein Mehrzweckprozessor und/oder ein Spezialprozessor. Durch die obige Diskussion gelehrte Prozesse können auch (als Alternative zu eine Maschine oder in Kombination mit einer Maschine) durch elektronische Schaltungen durchgeführt werden, die ausgelegt sind, die Prozesse (oder einen Teil davon) ohne die Ausführung von Programmcode durchzuführen.

[0056] Um Programmcode zu speichern, kann ein Fertigungsgegenstand verwendet werden. Ein Fertigungsgegenstand, der Programmcode speichert, kann, ohne jedoch darauf beschränkt zu sein, als ein oder mehrere Speicher (z.B. ein oder mehrere Flash-Speicher, Direktzugriffsspeicher (statisch, dynamisch oder anderweitig)), optische Disks, CD-ROMs, DVD ROMs, EPROMs, EEPROMs, magnetische oder optische Karten oder eine andere Art maschinenlesbarer Medien ausgeführt sein, die zum Speichern elektronischer Befehle geeignet ist. Auch kann Programmcode von einem entfernten Computer (z.B. einem Server) auf einen anfordernden Computer (z.B. einen Client) über Datensignale heruntergeladen werden, die in einem Ausbreitungsmedium ausgeführt sind (z.B. über eine Kommunikationsverbindung (z.B. eine Netzverbindung)).

[0057] Es ist beabsichtigt, dass der Begriff „Speicher“ in der hier verwendeten Weise alle flüchtigen Speichermedien einbezieht, wie z.B. dynamischen Direktzugriffsspeicher (Dynamic Random Access Memory, DRAM) und statischen RAM (SRAM) oder andere Arten von Speicher, die anderswo in dieser Anmeldung beschrieben sind. Computerausführbare Befehle können auf nichtflüchtigen Speichergeräten gespeichert sein, wie z.B. einer magnetischen Festplatte, einer optischen Disk, und sind typischerweise durch einen direkten Speicherzugriffprozess während Ausführung von Software durch einen Prozessor in den Speicher geschrieben worden. Ein Fachmann wird unmittelbar erkennen, dass der Begriff „maschinenlesbares Speichermedium“ jedwede Art flüchtigen oder nichtflüchtigen Speichergerätes beinhaltet, auf das durch einen Prozessor zugegriffen werden kann.

[0058] Die vorhergehenden ausführlichen Beschreibungen werden im Hinblick auf Algorithmen und symbolische Repräsentationen von Operationen an Datenbits innerhalb eines Computerspeichers dargelegt. Diese algorithmischen Beschreibungen und Repräsentationen sind die Hilfsmittel, die durch Fachleute im Datenverarbeitungswesen verwendet werden, um anderen Fachleuten die Substanz ihrer Ar-

beit am effektivsten zu vermitteln. Ein Algorithmus wird hier und im Allgemeinen als selbstkonsistente Abfolge von Operationen begriffen, die zu einem gewünschten Ergebnis führt. Die Operationen sind jene, die physikalische Manipulationen physikalischer Größen erfordern. Üblicher-, aber nicht notwendigerweise nehmen diese Größen die Form elektrischer oder magnetischer Signale an, die in der Lage sind, gespeichert, übertragen, kombiniert, verglichen und anderweitig manipuliert zu werden. Es hat sich hin und wieder als zweckdienlich erwiesen, vornehmlich aus Gründen der üblichen Verwendung diese Signale als Bits, Werte, Elemente, Symbole, Zeichen, Terme, Zahlen oder dergleichen zu bezeichnen.

[0059] Es muss jedoch beachtet werden, dass all diese und ähnliche Begriffe den entsprechenden physikalischen Größen zuzuordnen sind und hauptsächlich praktische Bezeichnungen sind, die auf diese Größen angewendet werden. Sofern nicht wie aus obiger Diskussion offensichtlich eigens anders erklärt, versteht es sich, dass in der gesamten Beschreibung Diskussionen, die Begriffe wie z.B. „Verarbeitung“ oder „Computing“ oder „Berechnen“ oder „Bestimmen“ oder „Anzeigen“ oder dergleichen nutzen, sich auf die Aktion und die Prozesse eines Computersystems oder ähnlichen elektronischen Computing-Gerätes beziehen, das Daten, die als physikalische (elektronische) Größen innerhalb der Register und Speicher des Computersystems repräsentiert sind, in andere Daten umarbeitet und transformiert, die in ähnlicher Weise als physikalische Größen innerhalb der Computersystemspeicher oder -register oder anderer derartiger Informationsspeicher-, -übertragungs- oder -anzeigergeräte repräsentiert sind.

[0060] Die beschriebenen Ausführungsformen betreffen auch eine Vorrichtung zum Durchführen der hierin beschriebenen Operationen. Diese Vorrichtung kann speziell für den gewünschten Zweck aufgebaut sein, oder sie kann einen Mehrzweckcomputer umfassen, der durch ein Computerprogramm, das im Computer gespeichert ist, selektiv aktiviert oder umkonfiguriert wird. So oder so stellt die Vorrichtung die Mittel zum Ausführen der hierin beschriebenen Operationen bereit. Das Computerprogramm kann in einem computerlesbaren Speichermedium gespeichert sein, wie z.B., ohne jedoch darauf beschränkt zu sein, jedweder Art von Disk einschließlich Floppy-Disks, optischer Disks, CD-ROMs und magneto-optischer Disks, Lesespeichern (Read-Only Memories, ROMs), RAMs, EPROMs, EEPROMs, magnetischer oder optischer Karten oder jedwedes anderen zum Speichern elektronischer Befehle geeigneten Medientyps, und wobei jedes davon mit einem Computersystembus gekoppelt ist.

[0061] Die hierin präsentierten Prozesse und Anzeigen sind nicht von Natur aus auf einen bestimmten Computer oder eine sonstige Vorrichtung bezogen.

Verschiedene Mehrzwecksysteme können mit Programmen gemäß den Ausführungen hierin benutzt werden, oder es kann sich als günstig erweisen, eine spezialisiertere Vorrichtung aufzubauen, um die beschriebenen Operationen durchzuführen. Die erforderliche Struktur für eine Vielzahl dieser Systeme wird aus der Beschreibung ersichtlich, die in dieser Anmeldung bereitgestellt ist. Darüber hinaus sind die Ausführungsformen nicht unter Bezug auf irgendeine bestimmte Programmiersprache beschrieben. Man wird verstehen, dass eine Vielzahl von Programmiersprachen verwendet werden könnte, um die Lehren der Ausführungsformen, wie hierin beschrieben, zu implementieren.

[0062] Zusätzliche Beispielimplementierungen sind wie folgt:

[0063] Ein Beispielverfahren, -system, eine Beispielvorrichtung oder ein beispielhaftes computerlesbares Medium für Objektverfolgung umfasst mindestens einen Kanal mehrerer Eingangseinzelbilder, einen Datensatz kontinuierlichen Lernens zum Akkumulieren von Mustern eines Zielobjektes, das in einem Eingangseinzelbild der mehreren Eingangseinzelbilder verfolgt wird, mindestens einen Prozessor und eine Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, um ein Zielobjekt zu erkennen, das mindestens ein Merkmal aufweist, ein Kandidatenobjekt in einem Eingangseinzelbild der mehreren Eingangseinzelbilder zu verfolgen und das Kandidatenobjekt als das Zielobjekt basierend auf dem mindestens einen Merkmal des Zielobjektes und einem vorher akkumulierten Muster des Zielobjektes zu identifizieren.

[0064] In einer anderen Beispielimplementierung dient die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu, einen Teil des Eingangseinzelbildes, der das Kandidatenobjekt enthält, das als das Zielobjekt identifiziert worden ist, im Datensatz kontinuierlichen Lernens zu akkumulieren, wobei der Teil ein Muster des Zielobjektes repräsentiert.

[0065] In einer anderen Beispielimplementierung dient die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu, einen Ort des Kandidatenobjektes, das als das Zielobjekt identifiziert worden ist, zu verfeinern, um den zu akkumulierenden Teil des Eingangseinzelbildes genau zu ermitteln, einschließlich des Durchführens irgendeines von einem Verfeinerungsnetz und Clustering des Kandidatenobjektes, das als das Zielobjekt identifiziert worden ist, und vorher akkumulierter Muster des Zielobjektes im Datensatz kontinuierlichen Lernens.

[0066] In einer anderen Beispielimplementierung dient, um das Kandidatenobjekt als das Zielobjekt ba-

sierend auf dem mindestens einen Merkmal des Zielobjektes und dem Muster des Zielobjektes zu identifizieren, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu, zu ermitteln, ob das mindestens eine Merkmal des Zielobjektes im Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, einen Zielabstand zwischen mindestens einem Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist, einen oder mehrere Musterabstände zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist, und zu ermitteln, dass ein Durchschnitt des berechneten Zielabstandes und der berechneten Musterabstände innerhalb der Identitätsschwelle liegt.

[0067] In einer anderen Beispielimplementierung dient die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu, basierend auf dem mindestens einen Merkmal, das im Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, das Kandidatenobjekt als nicht das Zielobjekt zu identifizieren, das Zielobjekt, das das mindestens eine Merkmal aufweist, wiederzuerkennen und das Kandidatenobjekt in dem Eingangseinzelbild der mehreren Eingangseinzelbilder unter Verwendung des wiedererkannten Zielobjektes zu verfolgen.

[0068] In einer anderen Beispielimplementierung dient, um basierend auf dem mindestens einen Merkmal, das im Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, das Kandidatenobjekt als nicht das Zielobjekt zu identifizieren, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu, einen oder mehrere Zielabstände zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist, einen oder mehrere Musterabstände zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist, und zu ermitteln, dass ein Durchschnitt der berechneten Zielabstände und Musterabstände die Identitätsschwelle überschreitet.

[0069] In einer anderen Beispielimplementierung beinhaltet das mindestens eine Merkmal, das in jedwedem des Kandidatenobjektes, des Zielobjektes und des Musters des Zielobjektes vorhanden ist, ein Erscheinungsbild des Objektes, das der Veränderung fähig ist, einschließlich jedwedem von einer Farbe, einer Form, einer Richtung und einem Bewegungsmerkmal.

[0070] In einer anderen Beispielimplementierung dient, um ein Kandidatenobjekt in einem Eingangseinzelbild der mehreren Eingangseinzelbilder zu ver-

folgen, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu, einen Klassifizierer auszuführen, um zwischen dem Kandidatenobjekt und anderen Objekten basierend auf dem mindestens einen Merkmal zu unterscheiden, das in jedwedem des Zielobjektes, des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist.

[0071] In der vorangehenden Spezifikation sind Ausführungsformen unter Bezug auf spezielle Ausführungsbeispiele oder -implementierungen beschrieben worden. Es ist offensichtlich, dass verschiedene Modifikationen an den beschriebenen Ausführungsformen oder Implementierungen vorgenommen werden können, ohne den weiteren Geist und Umfang der Ausführungsformen oder Implementierungen zu verlassen, wie in den nachfolgenden Ansprüchen dargelegt. Die Spezifikation und die Zeichnungen sind dementsprechend eher in einem veranschaulichenden Sinne als in einem beschränkenden Sinne zu betrachten.

Patentansprüche

1. Computerimplementiertes System zur Objektverfolgung, das Folgendes umfasst:
mindestens einen Kanal mehrerer Eingangseinzelbilder;
einen Datensatz kontinuierlichen Lernens zum Akkumulieren von Mustern eines Zielobjektes, das in einem Eingangseinzelbild der mehreren Eingangseinzelbilder verfolgt wird;
mindestens einen Prozessor und
eine Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, um ein Zielobjekt zu erkennen, das mindestens ein Merkmal aufweist,
ein Kandidatenobjekt in einem Eingangseinzelbild der mehreren Eingangseinzelbilder zu verfolgen und das Kandidatenobjekt als das Zielobjekt basierend auf dem mindestens einen Merkmal des Zielobjektes und einem vorher akkumulierten Muster des Zielobjektes zu identifizieren.

2. Computerimplementiertes System nach Anspruch 1, wobei die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, einen Teil des Eingangseinzelbildes, der das Kandidatenobjekt enthält, das als das Zielobjekt identifiziert worden ist, im Datensatz kontinuierlichen Lernens zu akkumulieren, wobei der Teil ein Muster des Zielobjektes repräsentiert.

3. Computerimplementiertes System nach irgendeinem der vorhergehenden Ansprüche 1 und 2, wobei die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, einen Ort des Kandidatenobjektes,

das als das Zielobjekt identifiziert worden ist, zu verfeinern, um den zu akkumulierenden Teil des Eingangseinzelbildes genau zu ermitteln, einschließlich des Durchführens irgendeines von einem Verfeinerungsnetz und Clustering des Kandidatenobjektes, das als das Zielobjekt identifiziert worden ist, und vorher akkumulierter Muster des Zielobjektes im Datensatz kontinuierlichen Lernens.

4. Computerimplementiertes System nach irgendeinem der vorhergehenden Ansprüche 1, 2 und 3, wobei, um das Kandidatenobjekt als das Zielobjekt basierend auf dem mindestens einen Merkmal des Zielobjektes und dem Muster des Zielobjektes zu identifizieren, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, zu ermitteln, ob das mindestens eine Merkmal des Zielobjektes in dem Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, einen Zielabstand zwischen mindestens einem Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist, einen oder mehrere Musterabstände zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist, und zu ermitteln, dass ein Durchschnitt des berechneten Zielabstandes und der berechneten Musterabstände innerhalb der Identitätsschwelle liegt.

5. Computerimplementiertes System nach irgendeinem der vorhergehenden Ansprüche 1, 2, 3 und 4, wobei die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, basierend auf dem mindestens einen Merkmal, das im Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, das Kandidatenobjekt als nicht das Zielobjekt zu identifizieren, das Zielobjekt wiederzuerkennen, das das mindestens eine Merkmal aufweist; und das Kandidatenobjekt im Eingangseinzelbild der mehreren Eingangseinzelbilder unter Verwendung des wiedererkannten Zielobjektes zu verfolgen.

6. Computerimplementiertes System nach irgendeinem der vorhergehenden Ansprüche 1, 2, 3, 4 und 5, wobei, um basierend auf dem mindestens einen Merkmal, das im Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, das Kandidatenobjekt als nicht das Zielobjekt zu identifizieren, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, einen oder mehrere Zielabstände zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist,

einen oder mehrere Musterabstände zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist, und zu ermitteln, dass ein Durchschnitt der berechneten Zielabstände und Musterabstände die Identitätsschwelle überschreitet.

7. Computerimplementiertes System nach irgendeinem der vorhergehenden Ansprüche 1, 2, 3, 4, 5 und 6, wobei das mindestens eine Merkmal, das in jedwedem des Kandidatenobjektes, des Zielobjektes und des Musters des Zielobjektes vorhanden ist, ein Erscheinungsbild des Objektes beinhaltet, das der Veränderung fähig ist, einschließlich jedwedem von einer Farbe, einer Form, einer Richtung und einem Bewegungsmerkmal.

8. Computerimplementiertes System nach irgendeinem vorhergehender Ansprüche 1, 2, 3, 4, 5, 6 und 7, wobei, um ein Kandidatenobjekt in einem Eingangseinzelbild der mehreren Eingangseinzelbilder zu verfolgen, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, einen Klassifizierer auszuführen, um zwischen dem Kandidatenobjekt und anderen Objekten basierend auf dem mindestens einen Merkmal zu unterscheiden, das in jedwedem des Zielobjektes, des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist.

9. Computerimplementiertes Verfahren zur Objektverfolgung, das Folgendes umfasst:
Erkennen eines Zielobjektes, das mindestens ein Merkmal aufweist,
Akkumulieren von Mustern des Zielobjektes, die in Eingangseinzelbildern einer Videosequenz identifiziert wurden;
Verfolgen eines Kandidatenobjektes in einem Eingangseinzelbild der Videosequenz; und
Identifizieren des Kandidatenobjektes als das Zielobjekt basierend auf dem mindestens einen Merkmal des Zielobjektes und einem oder mehreren akkumulierten Mustern des Zielobjektes.

10. Computerimplementiertes Verfahren nach Anspruch 9, das ferner Folgendes umfasst:
Akkumulieren eines Teils des Eingangseinzelbildes, der das Kandidatenobjekt enthält, das als das Zielobjekt identifiziert worden ist, in einem Datensatz kontinuierlichen Lernens, wobei der Teil ein Muster des Zielobjektes repräsentiert.

11. Computerimplementiertes Verfahren nach irgendeinem der vorhergehenden Ansprüche 9 und 10, das ferner Folgendes umfasst:
Verfeinern eines Ortes des Kandidatenobjektes, das als das Zielobjekt identifiziert worden ist, um den zu akkumulierenden Teil des Eingangseinzelbildes genau zu ermitteln, einschließlich des Durchführens ir-

gendeines von einem Verfeinerungsnetz und Clustering mit dem Kandidatenobjekt, das als das Zielobjekt identifiziert worden ist, und vorher akkumulierten Mustern des Zielobjektes im Datensatz kontinuierlichen Lernens.

12. Computerimplementiertes Verfahren nach irgendeinem der vorhergehenden Ansprüche 9, 10 und 11, wobei Identifizieren des Kandidatenobjektes als das Zielobjekt basierend auf dem mindestens einen Merkmal des Zielobjektes und dem einen oder den mehreren akkumulierten Mustern des Zielobjektes ferner Folgendes beinhaltet:

Berechnen, dass das mindestens eine Merkmal des Zielobjektes im Kandidatenobjekt und in dem einen oder den mehreren akkumulierten Mustern des Zielobjektes vorhanden ist,

Berechnen eines oder mehrerer Zielabstände zwischen mindestens einem Merkmal, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist, und

Ermitteln, dass ein Durchschnitt des einen oder der mehreren berechneten Ziel- und Musterabstände innerhalb der Identitätsschwelle liegt.

13. Computerimplementiertes Verfahren nach irgendeinem der vorhergehenden Ansprüche 9, 10, 11 und 12, das ferner Folgendes umfasst:

basierend auf dem mindestens einen Merkmal, das im Kandidatenobjekt und dem einen oder den mehreren akkumulierten Mustern des Zielobjektes vorhanden ist, Identifizieren des Kandidatenobjektes als nicht das Zielobjekt;

Wiedererkennen des Zielobjektes, das das mindestens eine Merkmal aufweist; und

Verfolgen des Kandidatenobjektes im Eingangseinzelbild der Videosequenz unter Verwendung des wiedererkannten Ziel Objektes.

14. Computerimplementiertes Verfahren nach irgendeinem der vorhergehenden Ansprüche 9, 10, 11, 12 und 13, wobei Identifizieren des Kandidatenobjektes als nicht das Zielobjekt basierend auf dem mindestens einen Merkmal, das im Kandidatenobjekt und dem einen oder den mehreren akkumulierten Mustern des Zielobjektes vorhanden ist, ferner Folgendes beinhaltet:

Berechnen eines oder mehrerer Zielabstände zwischen dem mindestens einen Merkmal, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist,

Berechnen eines oder mehrerer Musterabstände zwischen dem mindestens einen Merkmal, das in jedem des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist, und

Ermitteln, dass ein Durchschnitt des berechneten Zielabstandes und der berechneten Musterabstände die Identitätsschwelle überschreitet.

15. Computerimplementiertes Verfahren nach irgendeinem der vorhergehenden Ansprüche 9, 10, 11, 12, 13 und 14, wobei das mindestens ein Merkmal, das in jedwedem des Kandidatenobjektes, des Zielobjektes und des Musters des Zielobjektes vorhanden ist, ein Erscheinungsbild des Objektes beinhaltet, das der Veränderung fähig ist, einschließlich jedwedem von einer Farbe, einer Form, einer Richtung und einem Bewegungsmerkmal.

16. Computerimplementiertes Verfahren nach irgendeinem vorhergehender Ansprüche 9, 10, 11, 12, 13, 14 und 15, wobei Verfolgen eines Kandidatenobjektes in einem Eingangseinzelbild der Videosequenz ferner Folgendes beinhaltet:

Unterscheiden zwischen dem Kandidatenobjekt und anderen Objekten mithilfe eines Klassifizierers basierend auf dem mindestens einem Merkmal, das in jedwedem des Zielobjektes, des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist.

17. Computerlesbares Speichermedium oder computerlesbare Speichermedien, die mehrere Befehle umfassen, die bei Ausführung auf einem Prozessor veranlassen, dass der Prozessor ein Verfahren nach irgendeinem der Ansprüche 9-16 implementiert.

18. Vorrichtung zur Objektverfolgung, die Folgendes umfasst:

mindestens einen Speicher;
mindestens einen Prozessor, der kommunikativ mit dem Speicher gekoppelt ist,
mindestens einen Kanal von Einzelbildern einer Videosequenz;
mindestens einen Prozessor; und
eine Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, um ein Zielobjekt zu erkennen, das mindestens ein Merkmal aufweist,
ein Kandidatenobjekt in einem Eingangseinzelbild der Videosequenz zu verfolgen und
das Kandidatenobjekt als das Zielobjekt basierend auf dem mindestens einem Merkmal des Zielobjektes und einem Muster des Zielobjektes zu identifizieren, das vorher in dem mindestens einen Speicher akkumuliert worden ist.

19. Vorrichtung nach Anspruch 18, wobei die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, einen Teil des Eingangseinzelbildes, der das Kandidatenobjekt enthält, das als das Zielobjekt identifiziert worden ist, in dem mindestens einen Speicher zu akkumulieren, wobei der Teil ein Muster des Zielobjektes repräsentiert.

20. Vorrichtung nach irgendeinem der vorhergehenden Ansprüche 18 und 19, wobei die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, ei-

nen Ort des Kandidatenobjektes zu verfeinern, das als das Zielobjekt identifiziert worden ist, um den zu akkumulierenden Teil des Eingangseinzelbildes genau zu ermitteln, einschließlich irgendeines von einem Verfeinerungsnetz und Clustering mit dem Kandidatenobjekt, das als das Zielobjekt identifiziert worden ist, und vorher akkumulierten Mustern des Zielobjektes in dem mindestens einen Speicher durchzuführen.

21. Vorrichtung nach irgendeinem der vorhergehenden Ansprüche 18, 19 und 20, wobei, um das Kandidatenobjekt als das Zielobjekt basierend auf dem mindestens einem Merkmal des Zielobjektes und dem Muster des Zielobjektes zu identifizieren, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient,

zu ermitteln, ob das mindestens ein Merkmal des Zielobjektes in dem Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist,
einen Zielabstand zwischen mindestens einem Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist, und
einen Musterabstand zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist, und

zu ermitteln, dass ein Durchschnitt des berechneten Zielabstandes und Musterabstandes innerhalb der Identitätsschwelle liegt.

22. Vorrichtung nach irgendeinem der vorhergehenden Ansprüche 18, 19, 20 und 21, wobei die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient,

basierend auf dem mindestens einem Merkmal, das im Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, das Kandidatenobjekt als nicht das Zielobjekt zu identifizieren,
das Zielobjekt wiederzuerkennen, das das mindestens ein Merkmal aufweist; und
das Kandidatenobjekt im Eingangseinzelbild der Videosequenz unter Verwendung des wiedererkannten Zielobjektes zu verfolgen.

23. Vorrichtung nach irgendeinem der vorhergehenden Ansprüche 18, 19, 20, 21 und 22, wobei, um basierend auf dem mindestens einem Merkmal, das im Kandidatenobjekt und dem Muster des Zielobjektes vorhanden ist, das Kandidatenobjekt als nicht das Zielobjekt zu identifizieren, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient,
einen Zielabstand zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandidatenobjektes und des Zielobjektes vorhanden ist,
einen Musterabstand zwischen dem mindestens einen Merkmal zu berechnen, das in jedem des Kandi-

datenobjektes und des Musters des Zielobjektes vorhanden ist, und
zu ermitteln, ob ein Durchschnitt des berechneten Zielabstandes und Musterabstandes die Identitätsschwelle überschreitet.

24. Vorrichtung nach irgendeinem der vorhergehenden Ansprüche 18, 19, 20, 21, 22 und 23, wobei das mindestens ein Merkmal, das in jedwedem des Kandidatenobjektes, des Zielobjektes und des Musters des Zielobjektes vorhanden ist, ein Erscheinungsbild des Objektes beinhaltet, das der Veränderung fähig ist, einschließlich jedwedem von einer Farbe, einer Form, einer Richtung und einem Bewegungsmerkmal.

25. Vorrichtung nach irgendeinem der vorhergehenden Ansprüche 18, 19, 20, 21, 22, 23 und 24, wobei, um ein Kandidatenobjekt in einem Eingangseinzelbild der Videosequenz zu verfolgen, die Schaltung für kontinuierliches Lernen, die durch den mindestens einen Prozessor betrieben wird, ferner dazu dient, einen Klassifizierer auszuführen, um zwischen dem Kandidatenobjekt und anderen Objekten basierend auf dem mindestens einen Merkmal zu unterscheiden, das in jedwedem des Zielobjektes, des Kandidatenobjektes und des Musters des Zielobjektes vorhanden ist.

Es folgen 11 Seiten Zeichnungen

Anhängende Zeichnungen

100 Rahmenwerk für kontinuierliches Lernen zur Objektverfolgung

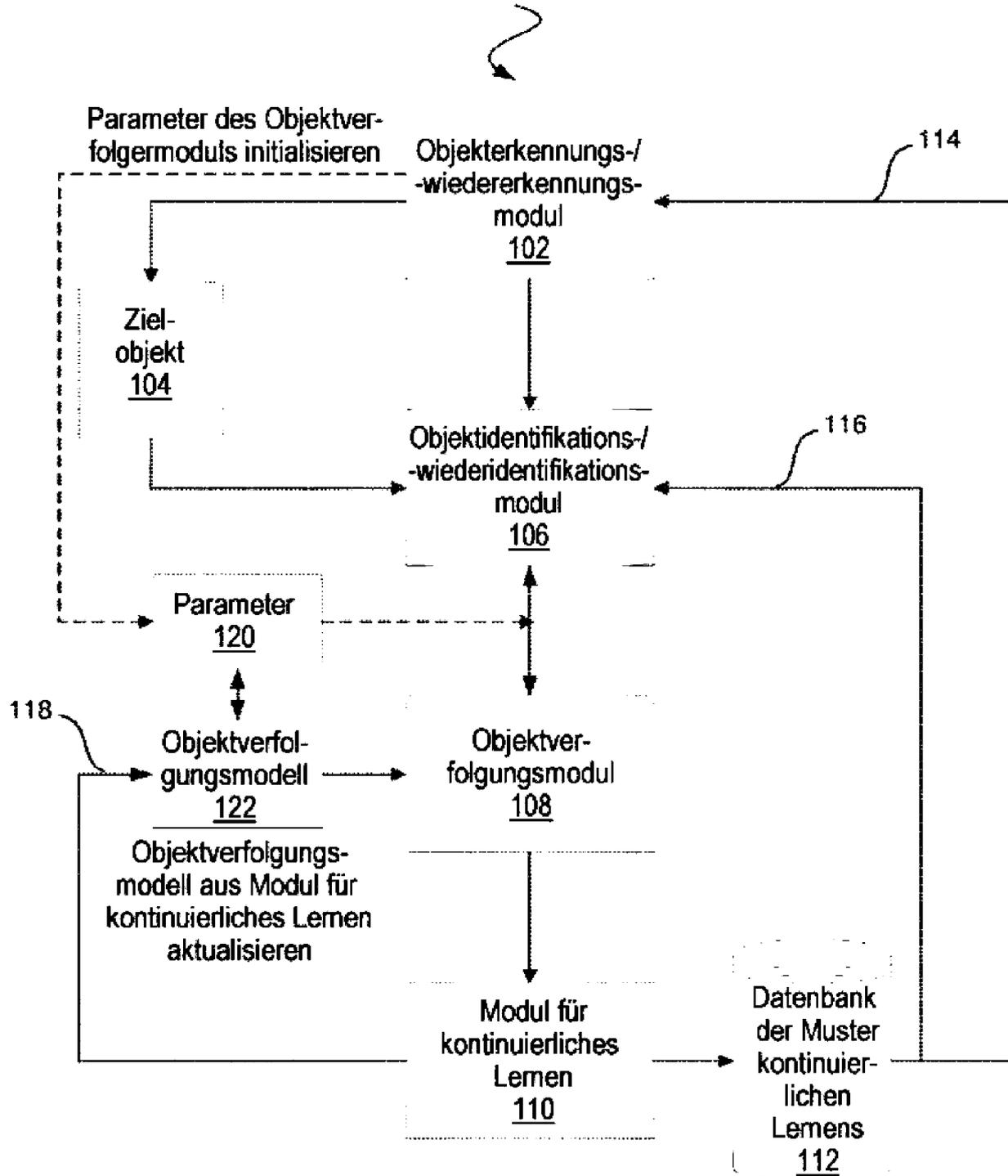


Fig.1

200 Architektur kontinuierlichen Lernens zur Objektverfolgung

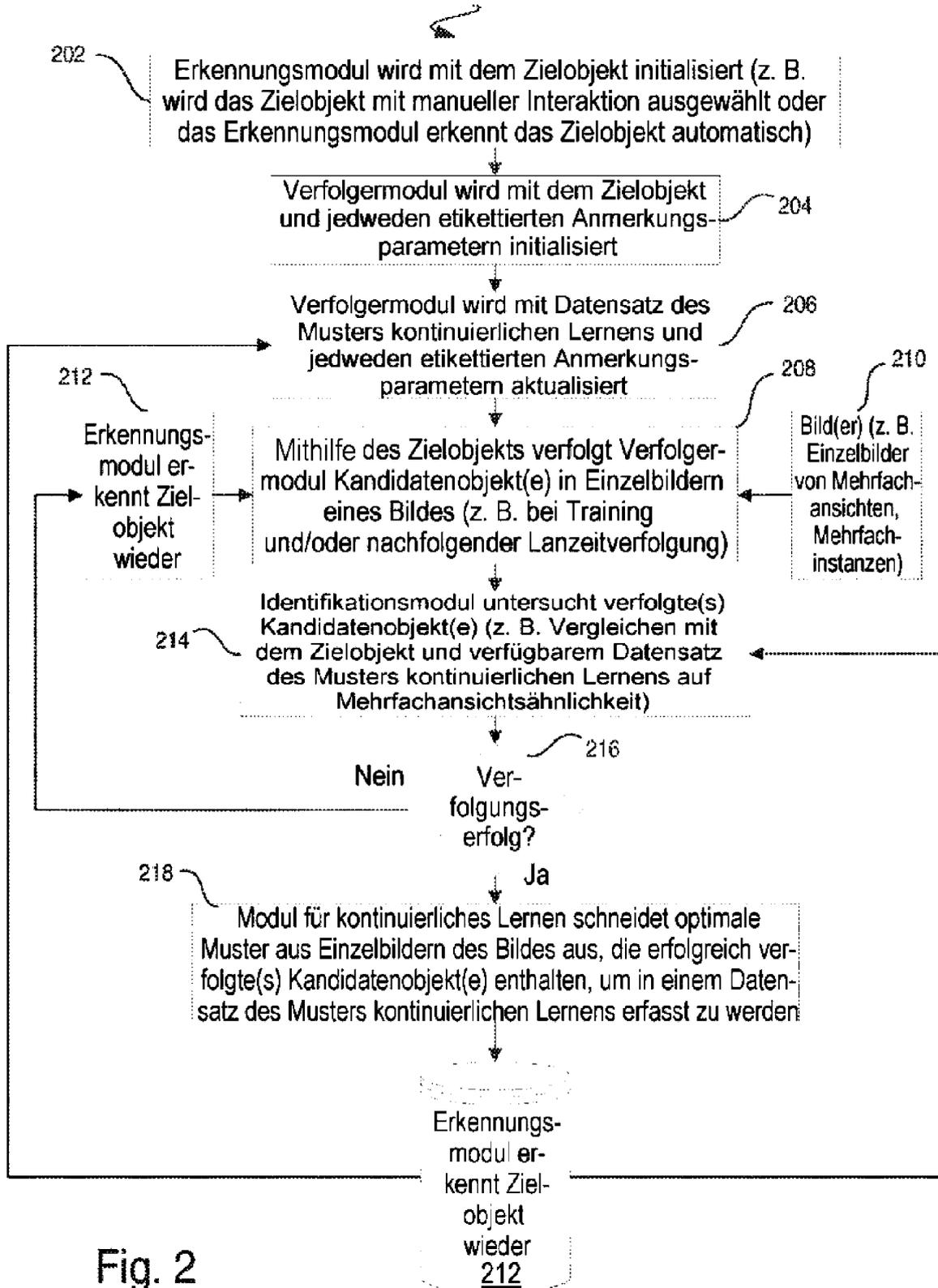


Fig. 2

300 Modul für kontinuierliches Lernen - Übersicht

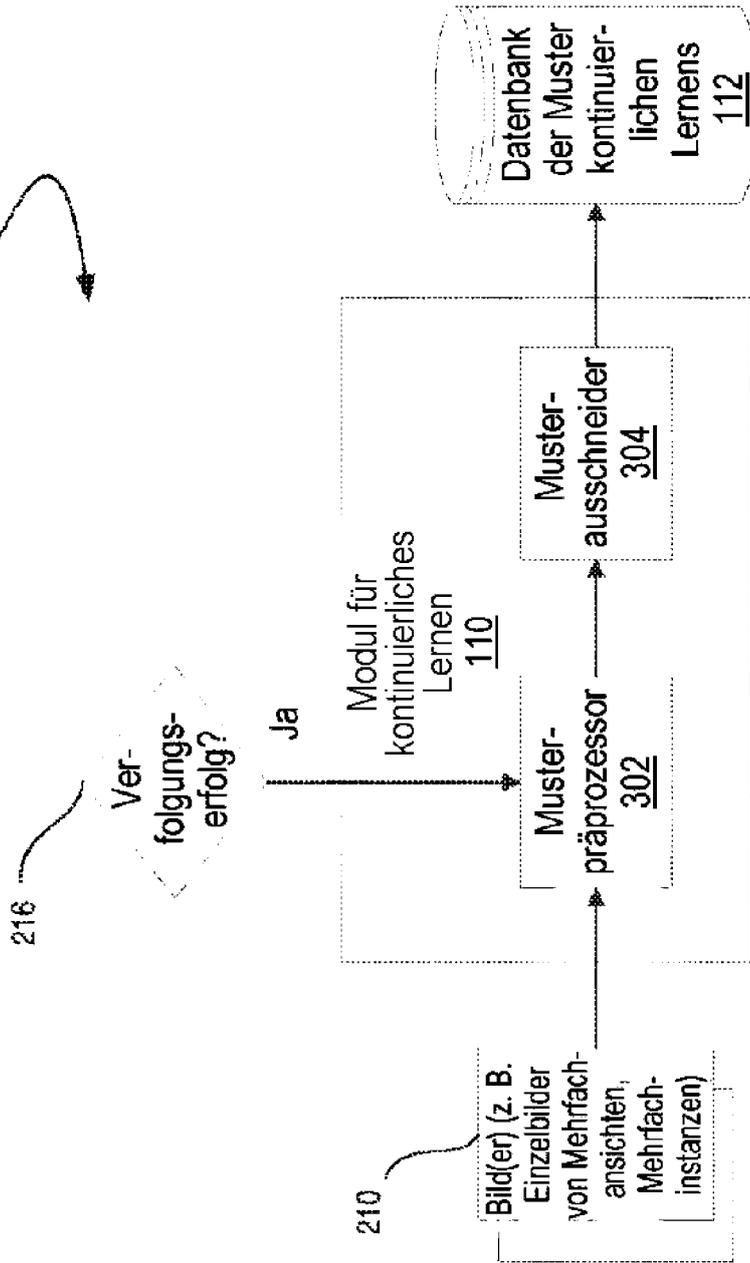


Fig. 3

400 - Beispiel für kontinuierliches Lernen

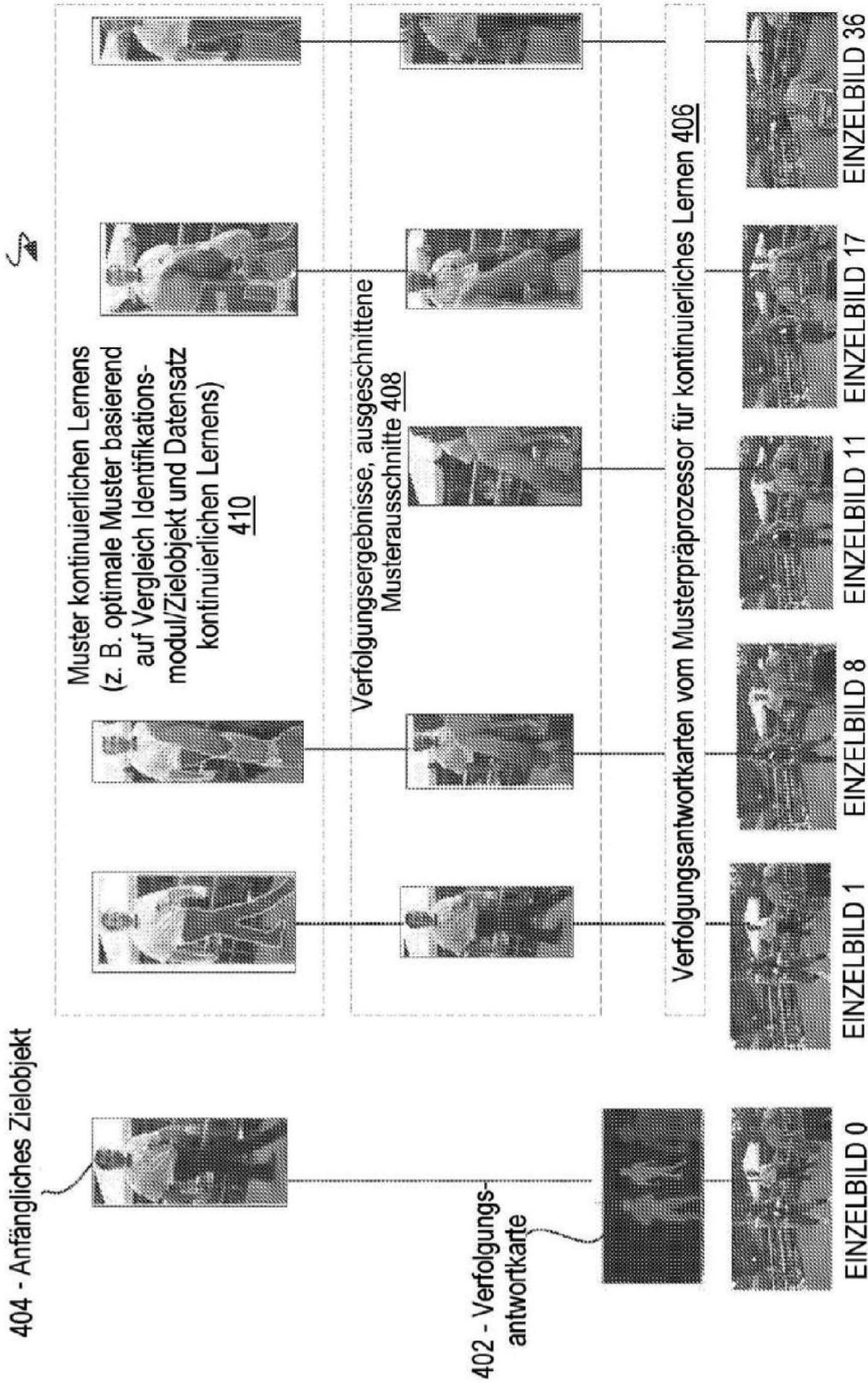


Fig. 4

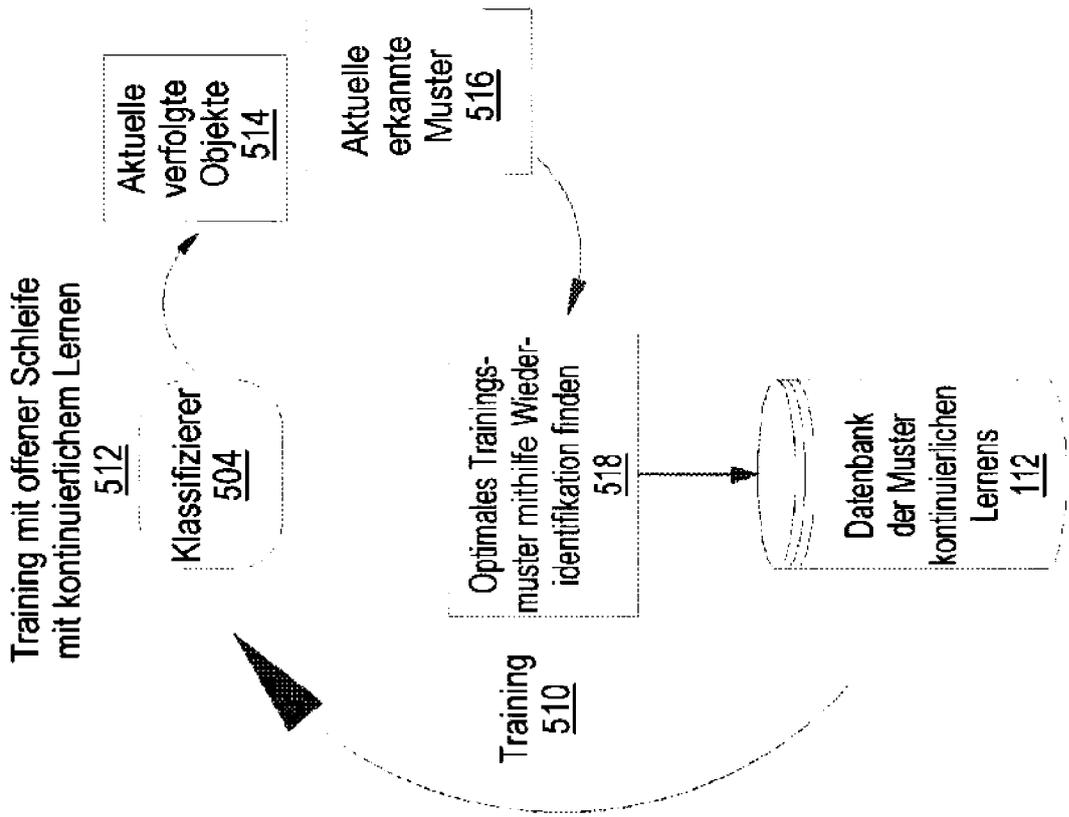


Fig.5b

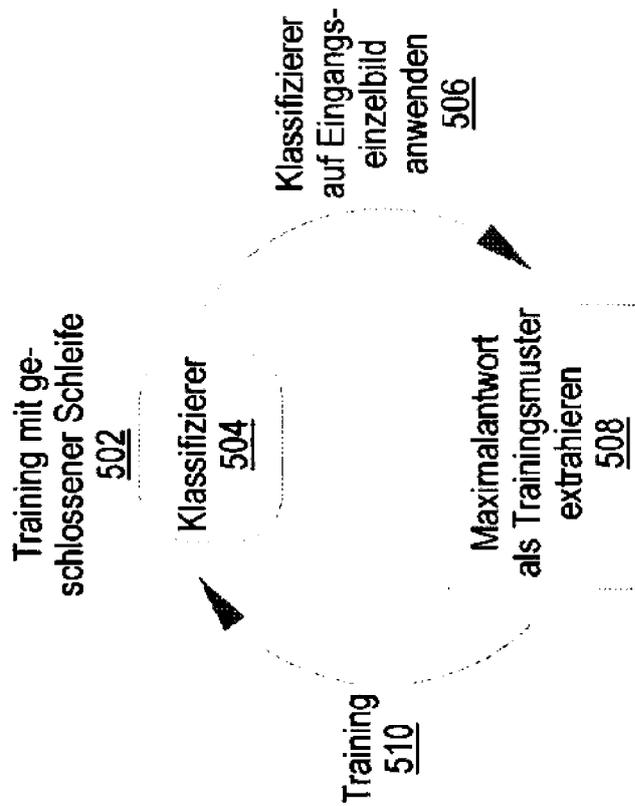


Fig.5a

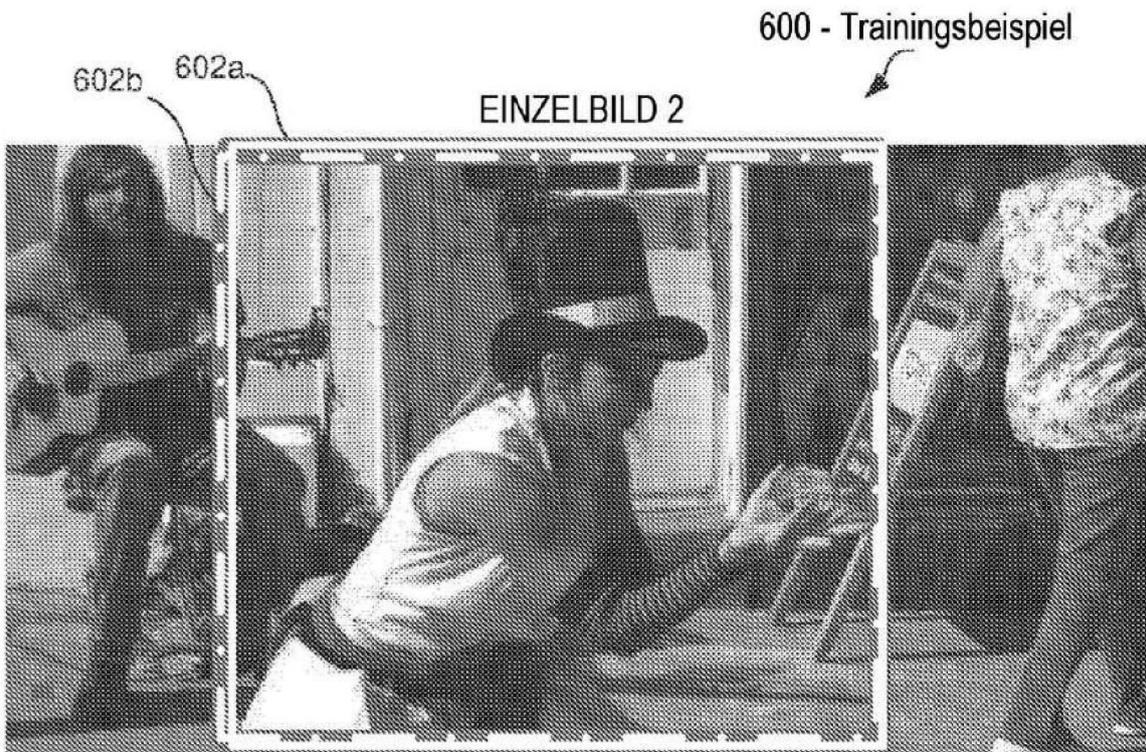


Fig. 6A

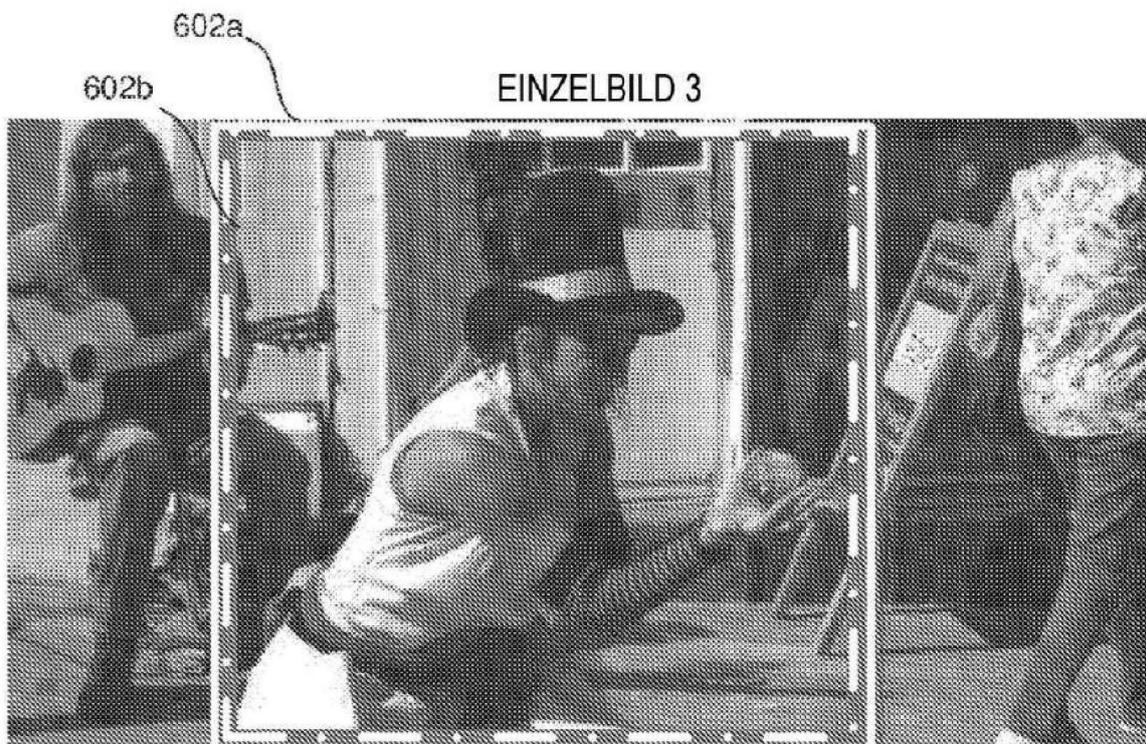


Fig. 6B

600 - Trainingsbeispiel (Fortsetzung)



Fig. 6C

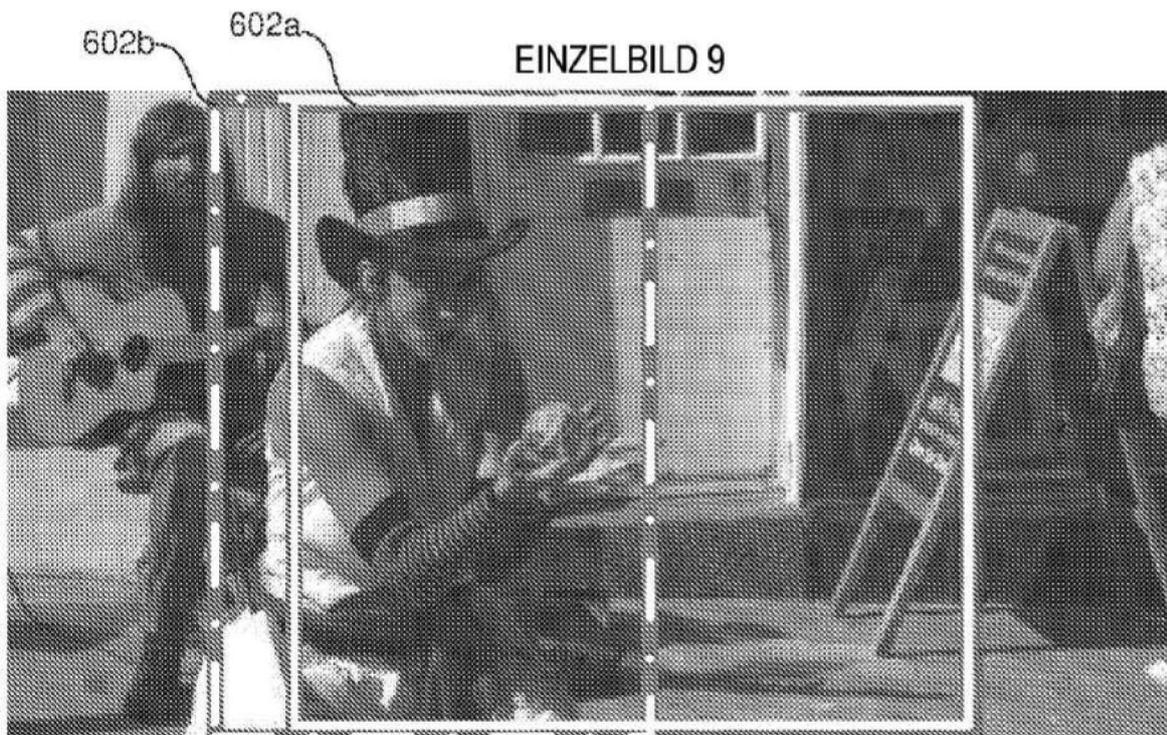


Fig. 6D

600 - Trainingsbeispiel (Fortsetzung)

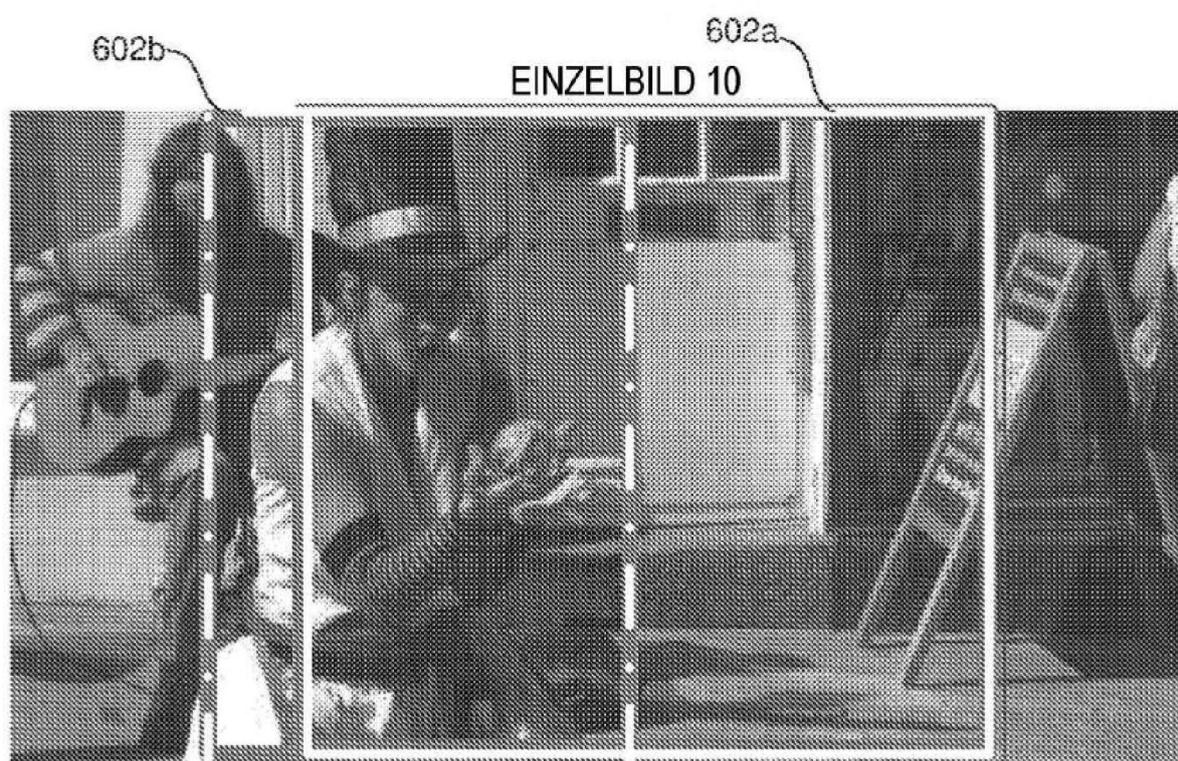


Fig. 6E

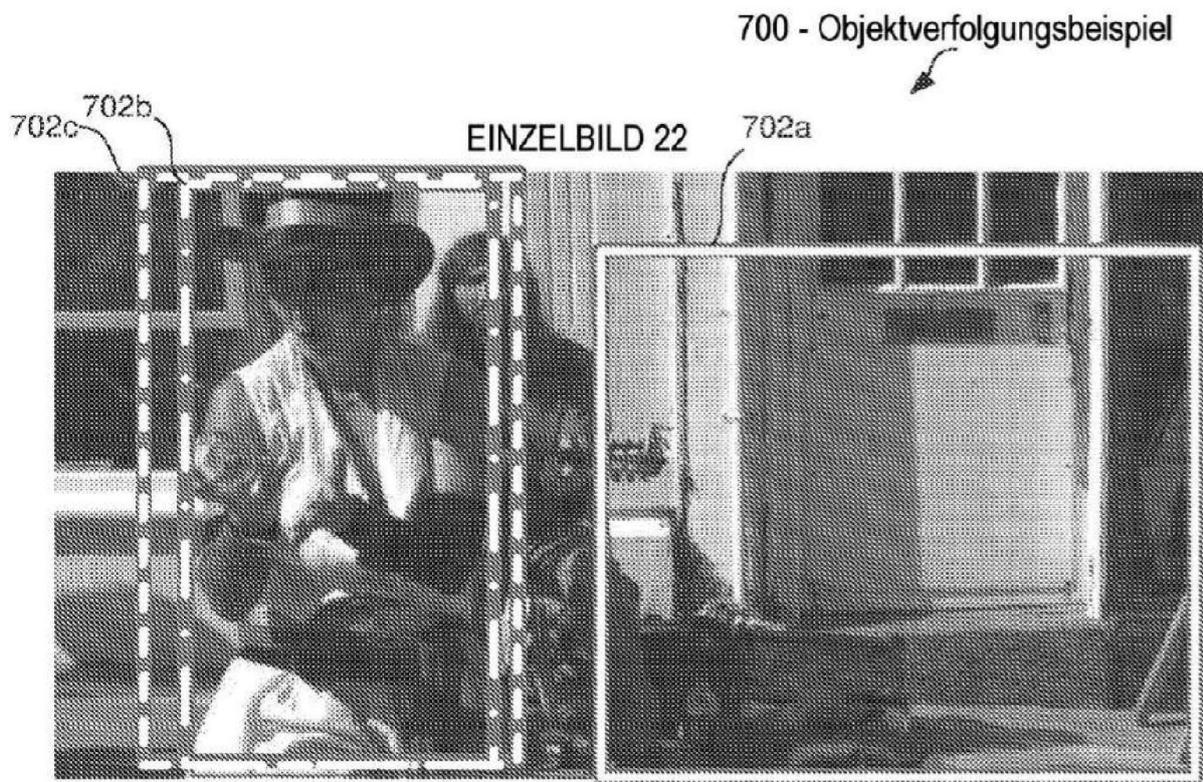


Fig. 7A

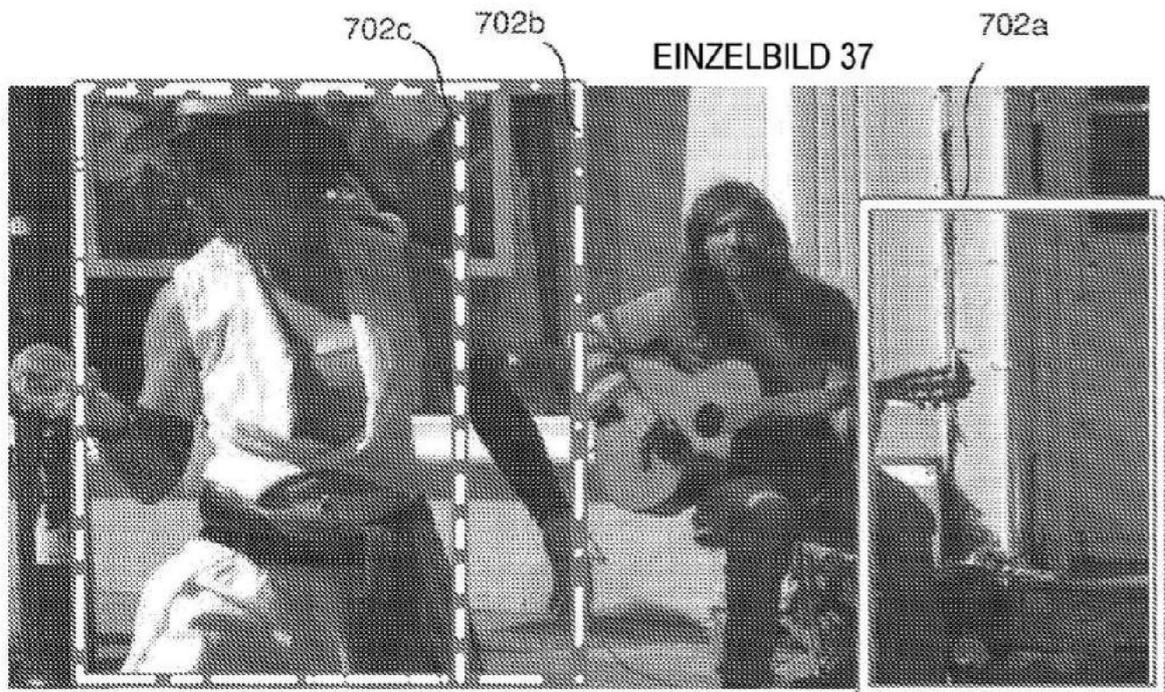


Fig. 7B

700 - Langzeit-Objektverfolgungsbeispiel (Fortsetzung)



Fig. 7C

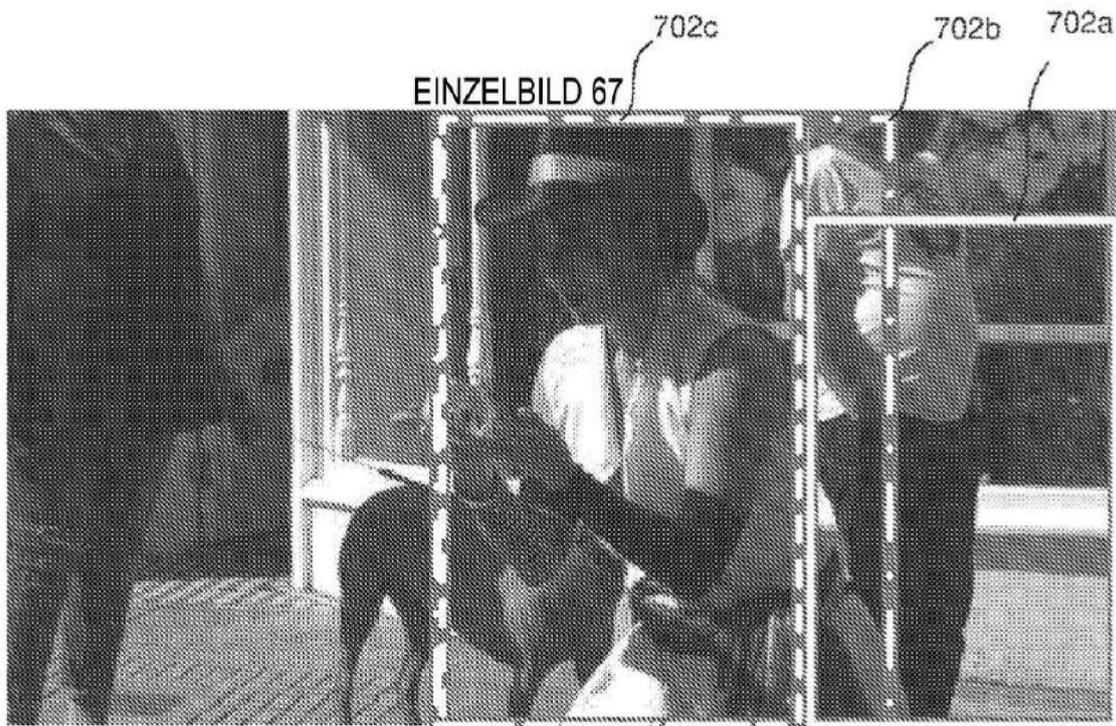


Fig. 7D

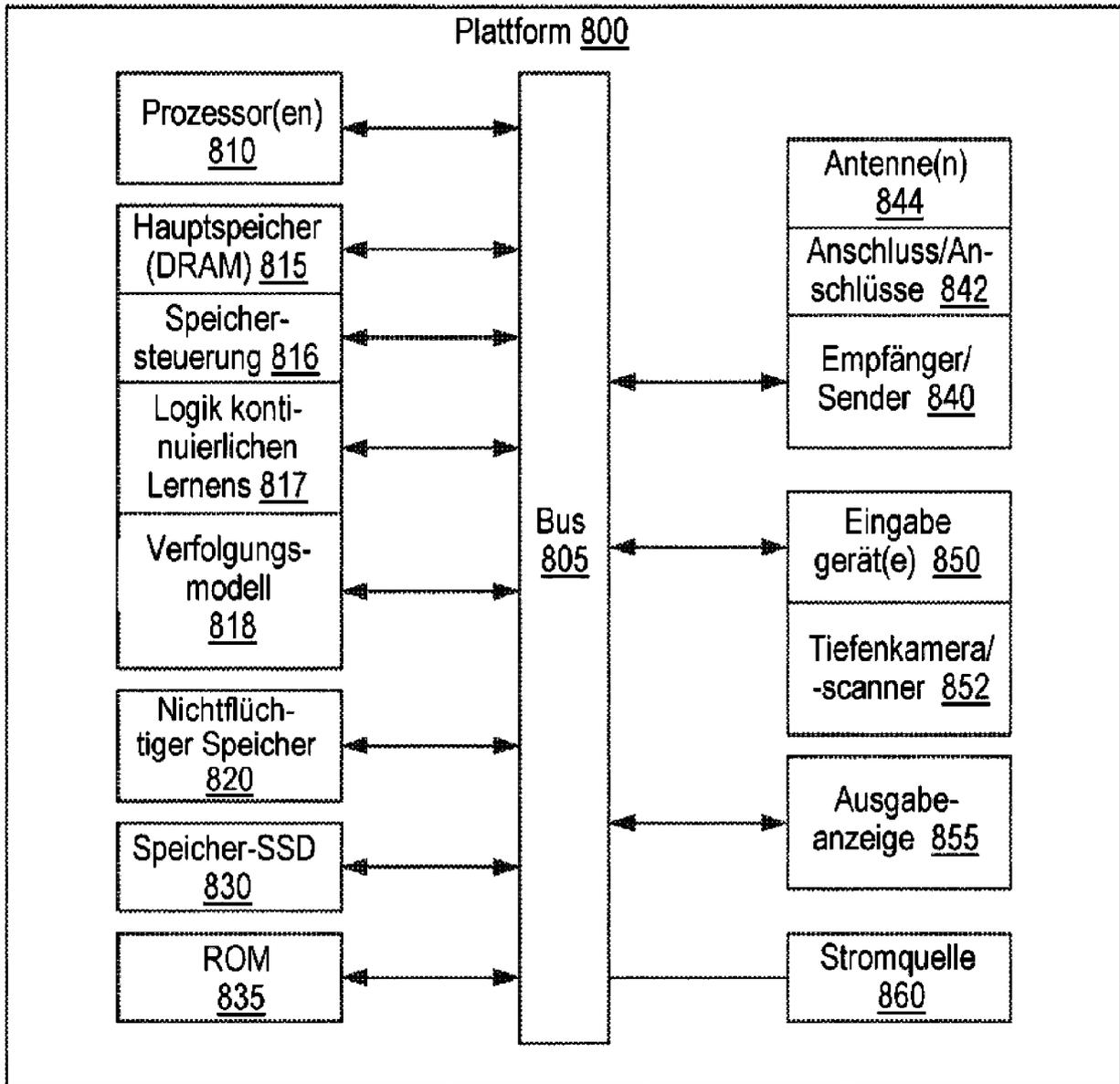


Fig.8