Precise Depth Image Based Real-Time 3D Difference Detection

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Dipl.-Inform. Svenja Kahn

geboren in München, Deutschland

Referenten der Arbeit:	Prof. Dr. techn. Dieter W. Fellner Technische Universität Darmstadt Prof. Dr. Didier Stricker Universität Kaiserslautern
Tag der Einreichung:	03.02.2014
Tag der mündlichen Prüfung:	25.03.2014

Erscheinungsjahr 2014
Darmstädter Dissertation
D 17

Abstract

3D difference detection is the task to verify whether the 3D geometry of a real object exactly corresponds to a 3D model of this object. Detecting differences between a real object and a 3D model of this object is for example required for industrial tasks such as prototyping, manufacturing and assembly control. State of the art approaches for 3D difference detection have the drawback that the difference detection is restricted to a single viewpoint from a static 3D position and that the differences cannot be detected in real time.

This thesis introduces real-time 3D difference detection with a hand-held depth camera. In contrast to previous works, with the proposed approach, geometric differences can be detected in real time and from arbitrary viewpoints. Therefore, the scan position of the 3D difference detection be changed on the fly, during the 3D scan. Thus, the user can move the scan position closer to the object to inspect details or to bypass occlusions.

The main research questions addressed by this thesis are:

Q1 How can 3D differences be detected in real time and from arbitrary viewpoints using a single depth camera?

Q2 Extending the first question, how can 3D differences be detected with a high precision?

Q3 Which accuracy can be achieved with concrete setups of the proposed concept for real time, depth image based 3D difference detection?

This thesis answers Q1 by introducing a real-time approach for depth image based 3D difference detection. The real-time difference detection is based on an algorithm which maps the 3D measurements of a depth camera onto an arbitrary 3D model in real time by fusing computer vision (depth imaging and pose estimation) with a computer graphics based analysis-by-synthesis approach.

Then, this thesis answers Q2 by providing solutions for enhancing the 3D difference detection accuracy, both by precise pose estimation and by reducing depth measurement noise. A precise variant of the 3D difference detection concept is proposed, which combines two main aspects. First, the precision of the depth camera's pose estimation is improved by coupling the depth camera with a very precise coordinate measuring machine. Second, measurement noise of the captured depth images is reduced and missing depth information is filled in by extending the 3D difference detection with 3D reconstruction.

The accuracy of the proposed 3D difference detection is quantified by a ground-truth based, quantitative evaluation. This provides an anwer to Q3. The accuracy is evaluated both for the basic setup and for the variants that focus on a high precision. The quantitative evaluation using real-world data covers both the accuracy which can be achieved with a time-of-flight camera (SwissRanger 4000) and with

a structured light depth camera (Kinect). With the basic setup and the structured light depth camera, differences of 8 to 24 millimeters can be detected from one meter measurement distance. With the enhancements proposed for precise 3D difference detection, differences of 4 to 12 millimeters can be detected from one meter measurement distance using the same depth camera.

By solving the challenges described by the three research question, this thesis provides a solution for precise real-time 3D difference detection based on depth images. With the approach proposed in this thesis, dense 3D differences can be detected in real time and from arbitrary viewpoints using a single depth camera. Furthermore, by coupling the depth camera with a coordinate measuring machine and by integrating 3D reconstruction in the 3D difference detection, 3D differences can be detected in real time and with a high precision.

Zusammenfassung

Bei einem 3D Soll-Ist Vergleich wird überprüft, ob die 3D Geometrie eines gegebenen Objektes exakt mit einem 3D Modell dieses Objektes übereinstimmt. Das Erkennen von Unterschieden zwischen einem realen Objekt und einem 3D Modell dieses Objektes wird unter anderem für verschiedene industrielle Szenarien benötigt. Beispiele hierfür sind Prototyping, Produktion und Fertigungskontrolle.

Bisherige Ansätze zum Erkennen von Unterschieden zwischen einem Objekt und einem 3D Modell des Objektes haben den Nachteil, dass die Differenzerkennung jeweils nur von einem einzelnen, statischen Blickpunkt aus vorgenommen werden kann. In der Regel werden hierfür hochpräzise Laser Scanner eingesetzt. Diese müssen allerdings nach jeder Repositionierung aufwendig neu eingemessen werden. Darüber hinaus können die vorliegenden Unterschiede mit bisherigen Ansätzen nicht in Echtzeit erfasst werden. Dadurch ist es nicht möglich, die Scan-Position während des Soll-Ist Abgleichs flexibel zu variieren, um beispielsweise 3D Unterschiede an einem anderen Bereich zu inspizieren oder um Verdeckungen zu umgehen.

Diese Dissertation stellt einen Echtzeit 3D Soll-Ist Vergleich mit einer Tiefenkamera vor. Im Gegensatz zu bisherigen Ansätzen können geometrische Unterschiede damit in Echtzeit und von beliebigen Blickpunkten aus erfasst werden. Der Benutzer kann dabei die Betrachungsposition frei wählen und zur Laufzeit beliebig verändern. Durch eine Repositionierung der Tiefenkamera während des Soll-Ist Abgleichs können Verdeckungen umgangen, vorliegende Unterschiede aus verschiedenen Perspektiven betrachtet und Details durch eine Bewegung der Kamera näher zum jeweils relevanten Objekt inspiziert werden.

Die wesentlichen Forschungsfragen dieser Arbeit lauten: Wie können 3D Differenzen mit einer Tiefenkamera in Echtzeit, von frei wählbaren Blickpunkten aus und mit einer hohen Genauigkeit erkannt werden? Wie können die 3D Messungen der Tiefenkamera dabei in Echtzeit den der echten Geometrie entsprechenden Punkten auf dem 3D-Modell zugeordnet werden? Durch welche Einflussfaktoren wird die Genauigkeit des 3D Soll-Ist Abgleiches bestimmt? Wie kann die Genauigkeit der 3D Differenzerkennung unter Berücksichtigung dieser Einflussfaktoren verbessert werden? Welche Genauigkeit wird hierbei insgesamt erreicht?

Zur Beantwortung dieser Fragen stellt diese Arbeit zunächst ein Konzept zur tiefenbildbasierten 3D Differenzerkennung vor. Eine wichtige Komponente dieses Konzeptes ist ein Algorithmus, welcher jedem 3D Messpunkt der Tiefenkamera einen 3D Punkt auf der Oberfläche des 3D Modells zuordnet. Dieser Algorithmus ordnet den 3D Messpunkten der Tiefenkamera auf dem echten Objekt 3D Punkte auf dem virtuellen 3D Modell zu, deren Lage denjenigen der Messpunkte auf dem echten Objekt entspricht. Die Echtzeitfähigkeit dieses Algorithmus wird durch die Kombination von Computer Vision mit einem Computer Graphik basiertem „Analyse durch Synthese“ Verfahren ermöglicht.

Darüber hinaus umfasst diese Arbeit eine Darstellung verschiedener Fehlerquellen, welche die Genauigkeit eine tiefenbildbasierten 3D Differenzerkennung einschränken. Die beiden Hauptquellen sind Ungenauigkeiten bei der Schätzung der Kamerapose sowie Ungenauigkeiten bei den von der Tiefenkamera erfassten Distanzwerten. Aus diesem Grund werden verschiedene Varianten zur Bestimmung der Pose der Tiefenkamera sowie zur Genauigkeitsverbesserung der erfassten 3D Messwerte vorgestellt und diskutiert. Darauf basierend wird eine Variante des 3D Soll-Ist Abgleichs entworfen, welche auf eine hohe Präzision ausgerichtet ist. Diese basiert zum einen auf einer sehr präzisen Bestimmung der Position und Orientierung der Tiefenkamera durch eine Kombination der Tiefenkamera mit einem portablen Messarm. Zum anderen wird das Rauschen der von der Tiefenkamera erfassten 3D Messwerte reduziert, indem ein Algorithmus zur 3D Oberflächenrekonstruktion in die 3D Differenzerkennung integriert wird.

Die Genauigkeit der 3D Differenzerkennung wird durch eine quantitative Evaluierung anhand aufgenommener Sequenzen evaluiert. Dabei wird sowohl die erreichbare Genauigkeit mit dem grundlegenden Setup untersucht als auch die Genauigkeit, welche mit der auf Präzision ausgerichteten Variante erreicht wird. Die Evaluierung wird sowohl für eine Time-of-Flight Tiefenkamera (SwissRanger 4000) als auch für eine Tiefenkamera durchgeführt, welche 3D Messwerte durch ein „Structured Light“ Verfahren anhand von projeziertem, strukturierten Licht bestimmt (Kinect). Mit dem grundlegenden Setup und der letztgenannten Tiefenkamera können Unterschiede erkannt werden, die (abhängig von der Messdistanz) mindestens 8 bis 24 Millimeter betragen. Mit dem Setup, welches auf eine hohe Präzision ausgerichtet ist, können dagegen bereits Unterschiede ab einer Abweichung von 4 bis 12 Millimetern erkannt werden.

Im Folgenden werden die wichtigsten Aspekte der einzelnen Kapitel dieser Dissertation zusammengefasst.

Hintergrund

Tiefenkameras Tiefenkameras erfassen dichte 3D Messungen in Echtzeit. Sie messen oder berechnen die Distanz eines erfassten Objektes zum Kamerazentrum an jedem 2D Pixel des Bildsensors. Zur Zeit sind die beiden am weitesten entwickelten Ansätze zur Echtzeit-Tiefenbilderfassung sogenannte "Time-of-Flight"Kameras und ein Verfahren zur Tiefenbilderfassung anhand von strukturiertem Licht (ßtructured light"). Beim erstgenannten Verfahren wird Licht von der Kamera emittiert. Dieses wird von der erfassten Szene reflektiert und vom Sensor der Kamera erfasst. Anhand des Zeitintervalls, das zwischen der Emission und der Erfassung des reflektierten Lichts vergangen ist, kann die Distanz zu den erfassten Objekten berechnet werden. Im Gegensatz dazu wird beim strukturierten Licht Verfahren ein spezielles Muster auf die Szene projiziert und von einer Kamera erfasst. Hierbei werden die Tiefeninformationen durch eine Analyse der Verzerrung des projizierten Musters gewonnen. Das in dieser Arbeit vorgestellte Konzept ist nicht auf diese beiden Verfahren zur Echtzeit-Tiefenbilderfassung beschränkt. Der Ansatz zur Echtzeit 3D Differenzerkennung, welcher in dieser Arbeit vorgestellt wird, kann für jedes 3D Messsystem eingesetzt werden, das dichte Tiefenbilder in Echtzeit erfasst.

Bisherige Ansätze Die bisherigen Ansätze zur Differenzerkennung können nicht für Echtzeit 3D Differenzerkennung mit einer beliebig bewegbaren Kamera eingesetzt werden, da keiner dieser Ansätze alle dafür nötigen Voraussetzungen erfüllt. Sie basieren in der Regel auf 3D Laser Scan Messungen oder auf der Erfassung von Photos oder Videos mit einer 2D Kamera. Aufgrund der folgenden Einschränken können bisherige Ansätze nicht zur Echtzeit 3D Differenzerkennung aus beliebigen Positionen eingesetzt werden:

- Der Ansatz von Webel ermöglicht die Erfassung von 3D Differenzen an einzelnen 3D Punkten, jedoch keine dichte 3D Differenzerfassung [WBSW07].
- Andere Ansätze erfassen keinerlei 3D Messungen [GSB*07] [SS08] [GBSN09] [FG11]. Bei diesen Ansätzen wird stattdessen ein 2D Bild des echten Objektes visuell mit dem 3D Modell überlagert. Die Differenzerkennung obliegt in diesem Fall dem Benutzer, durch einen manuellen, visuellen Vergleich des 3D Modells und des 2D Bildes.
- Der Ansatz von Tang et al. beschränkt die Differenzerkennung auf Abweichungen von einer einzelnen planaren Fläche [TAH09].
- Die meisten bisherigen Ansätze sind auf eine statische 3D-Erfassungsposition beschränkt [BTHC06] [ABG*06] [GSB*07] [Bos08] [GBSN09] [TAH09] [Bos10] [FG11]. Hierfür wird in der Regel ein statischer Laserscanner eingesetzt. Eine manuelle Neukalibrierung ist jedes Mal nötig, wenn die Position des Scanners verändert wurde, etwa durch das manuelle Auswählen korrespondierender Punkte in den erfassten 3D Daten und auf dem 3D Modell. Daher sind diese Ansätze nicht für bewegliche Kamerapositionen geeignet. Der Ansatz von Tang [TAAH11] setzt darüber hinaus voraus, dass das 3D Modell bereits mit den vom Laser Scanner erfassten Daten in einem gemeinsamen Koordinatensystem vorliegen muss.
- Die meisten bisherigen Ansätze sind nicht echtzeitfähig [ABG*06] [AML07] [GSB*07] [Bos08] [GBSN09] [TAH09] [NDB*10] [Bos10] [TAAH11] [FG11] [VDC12].

Konzept: Tiefenbildbasierte, echtzeitfähige 3D Differenzerkennung

Hinsichtlich eines echtzeitfähigen Vergleichs von 3D Messungen (die mit einer Tiefenkamera erfasst wurden) mit einem beliebigen 3D Modell bestehen zwei wesentliche Herausforderungen: Zum einen kann die Position der handgeführten Tiefenkamera beliebig vom Benutzer geändert werden und ändert sich für jedes erfasste Tiefenbild. Daher muss das Koordinatensystem der Tiefenkamera für jedes erfasste Tiefenbild neu in Bezug zum Koordinatensystem des 3D Modells gesetzt werden. Zum anderen müssen auch dann, wenn das Koordinatensystem der Tiefenkamera mit dem Koordinatensystem des 3D Modells in Übereinsteimmung gebracht wurde, noch 3D-3D Korrespondenzen zwischen den 3D Messungen und den entsprechenden 3D Punkten auf der Oberfläche des 3D Modells bestimmt werden ("welcher 3D Punkt auf der Oberfläche des 3D Modells entspricht einer gegebenen 3D Messung?"). Die geforderte Echtzeitfähigkeit stellt hierbei eine besondere Herausforderung dar, da das Tiefenbild mehrere hunderttausend Werte und das 3D Modell Millionen von Dreiecke umfassen kann.

Um einen echtzeitfähigen 3D Soll-Ist Abgleich zu ermöglichen, wird daher in dieser Arbeit ein Verfahren vorgestellt, welches auf der Kombination von ***Computer Vision*** und ***Computer Graphik*** beruht.

Computer Vision wird hierbei zur Verarbeitung der erfassten Tiefenbilder und zur Bestimmung der Pose der Tiefenkamera eingesetzt. Hierdurch wird das Koordinatensystem der Tiefenkamera mit dem Koordinatensystem des 3D Modells in übereinstimmung gebracht. Durch eine Aufteilung der für die Bestimmung der Kamerapose benötigten Registrierung in einen Offline-Schritt (Bestimmung der relativen Transformation zwischen einem Tracking-Gerät und der Tiefenkamera sowie zwischen dem 3D Modell und dem Tracking- Koordinatensystem) und einer zur Laufzeit ausgeführten Komponente (Bestimmung der Pose der Tiefenkamera) wird eine Echtzeitfähigkeit der Registrierung auch für variable Kameraposen ermöglicht.

Anschließend ermöglicht ein **Computer Graphik** basiertes Analyse-durch-Synthese Verfahren die echtzeitfähige Zuordnung der 3D Messungen zu entsprechenden Punkten auf der Oberfläche des 3D Modells. Hierfür wird das 3D Modell aus dem Blickwinkel der berechneten Kameraposition so gerendert, dass die Projektionsparameter des Renderings exakt den Abbildungseigenschaften der echten Tiefenkamera entsprechen. Daraufhin wird der Tiefenbuffer der Graphikkarte ausgelesen und Pixel für Pixel mit dem von der Tiefenkamera erfassten Tiefenbild verglichen.

Der vorgeschlagene Ansatz nutzt die massive Parallelisierung beim Rendering eines 3D Modells auf der Graphikkarte. Dadurch können 3D Messwerte in sehr kurzer Berechnungszeit entsprechenden 3D Punkten auf dem 3D Modell zugeordnet werden. Für 307.200 Messwerte und ein 3D Modell mit 2,5 Millionen Dreiecken können alle Differenzen etwa in weniger als 15 Millisekunden berechnet und visualisiert werden. Darüber hinaus kann der vorgeschlagene Ansatz auf beliebige 3D Modelle angewandt werden, welche gerendert werden können. Die interne Repräsentation der 3D Daten ist dabei beliebig, abgesehen vom Rendering des 3D Modells muss nicht auf diese zugegriffen werden. Daher kann das 3D Modell in einer beliebigen (solange renderbaren) Form vorliegen.

Anzahl der Dreiecke	**Bildgröße**		
	176×144	240×320	480×640
1.280	1ms	3ms	9ms
15.000	1ms	3ms	9ms
111.000	2ms	4ms	9ms
670.000	3ms	6ms	12ms
1.000.000	3ms	6ms	13ms
2.500.000	6ms	8ms	14ms

Tabelle 0.1.: Analysis-by-Synthesis Algorithmus zur Bestimmung von 3D-3D Korrespondenzen zw. erfassten Messwerten und 3D Punkten auf dem 3D Modell: Ausführungsdauer.

Tabelle 0.1 stellt die Ausführungsdauer der 3D Differenzerkennung (Bestimmung von 3D-3D Korrespondenzen sowie Berechnung und Visualisierung der Differenzen) in Abhängigkeit von Bildgröße

und Komplexität des 3D Modells dar. Die benötigte Rechendauer wurde auf einem Intel Core i7 mit einer GeForce GTX 470 Graphikkarte bestimmt. Hierbei handelt es sich um obere Grenzen der benötigten Laufzeit, da die angegebenen Zeiten einen optionalen Projektionsschritt der 3D-Differenzen auf ein zusätzliches Bild enthalten. Ohne diesen optionalen Zusatzschritt liegt die Ausführungsdauer für das $640 \cdot 480$ Bild jeweils ca. 3-4ms unter den angegebenen Werten.

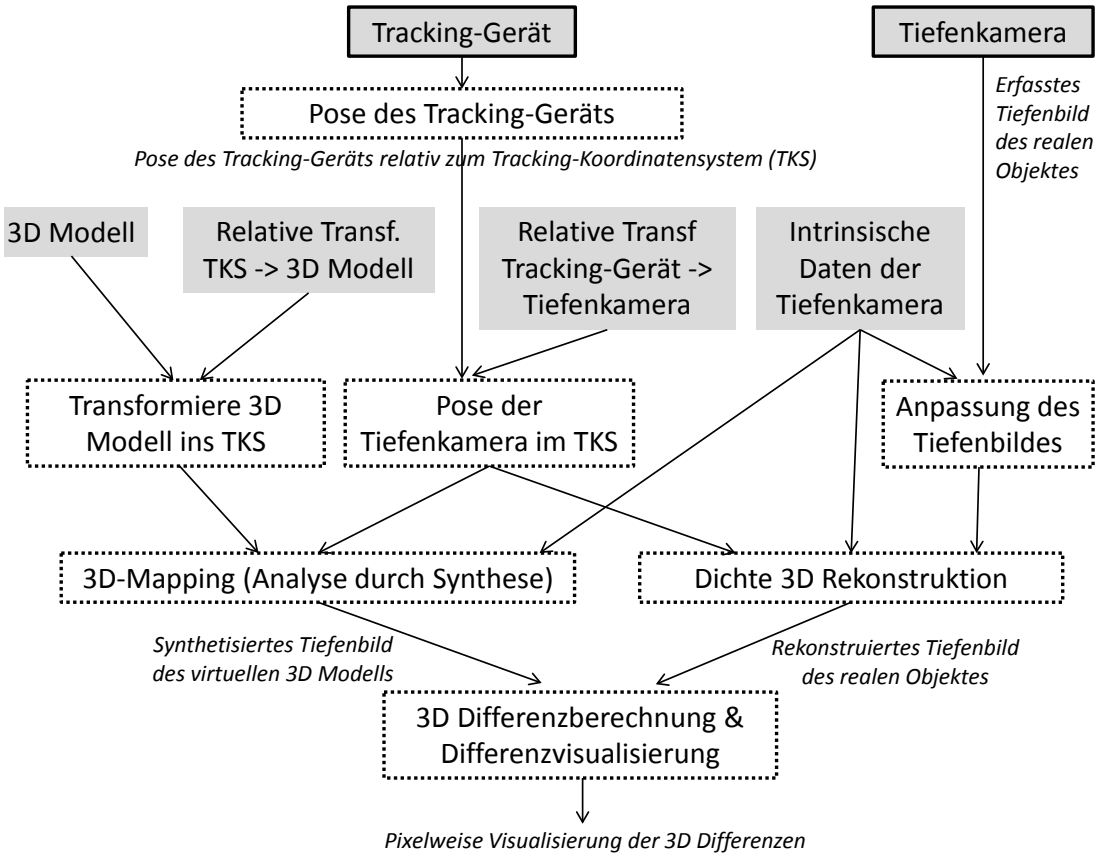


Abbildung 0.1.: Komponenten und Datenfluss der 3D Differenzerkennung.

Abbildung 0.1 stellt die algorithmischen Komponenten sowie den Datenfluss des allgemeinen Ansatzes zur 3D Differenzerkennung vor, der in dieser Arbeit vorgeschlagen wird. Die Position und Orientierung der Tiefenkamera wird anhand eines Tracking-Gerätes bestimmt. Hierbei kann es sich etwa um eine zusätzliche 2D Kamera handeln (deren Bild für eine bildbasierte Bestimmung der Kamerapose genutzt wird), um einen Roboterarm oder eine Koordinaten-Messmaschine (wie etwa einen portablen Messarm), oder um die Tiefenkamera selbst.

Anhand der in einem Vorbereitungsschritt berechneten relativen Transformation zwischen dem Tracking-Gerät und der Tiefenkamera sowie der relativen Transformation zwischen dem 3D Modell und dem Tracking-Koordinatensystem können sowohl die Pose der Tiefenkamera im Trackingkoordinatensystem als auch die Transformation des 3D Modells in dieses Koordinatensystem berechnet werden. Durch das beschriebene Analyse-durch-Synthese Verfahrens wird ein synthetisches Tiefenbild des 3D Modells bestimmt, dessen Distanzwerte den von der Tiefenkamera gemessenen Distanzwerten an der entsprechenden Pixelposition entsprechen.

Da die von Tiefenkameras erfassten Distanzmessungen sowohl Rauschen als auch systematischen Messfehlern unterliegen und teilweise fehlende Daten aufweisen (etwa, wenn in Teilbereichen des Tiefenbildes keine Distanzen erfasst werden konnten), wird der 3D Soll-Ist Abgleich um einen 3D Rekonstruktionsschritt ergänzt. Hierbei werden die 3D-Daten mehrer Tiefenbilder kombiniert, indem während des 3D Soll-Ist Abgleichs eine dichte 3D Rekonstruktion der erfassten Szene durchgeführt wird. Dadurch werden sowohl Messungenauigkeiten ausgeglichen als auch fehlende Daten in den Tiefenbildern ergänzt. Durch eine stark parallelisierte Implementierung auf der Graphikkarte wird eine dichte 3D Rekonstruktion in Echtzeit ermöglicht (die Ausführungsdauer liegt auf einer GeForce GTX 470 bei ca. 40 ms). Das von der Tiefenkamera erfasste Tiefenbild wird in diesem Fall bei der Differenzberechnung durch ein Tiefenbild ersetzt, das aus dem rekonstruierten 3D Modell extrahiert wurde.

Die nächsten beiden Kapitel dieser Arbeit behandeln die Frage, wie ein möglichst exakter 3D Soll-Ist Abgleich ermöglicht werden kann. Die beiden wesentlichen Fehlerquellen bei dem vorgestellten Ansatz ergeben sich zum einen aus Ungenauigkeiten der Position und Orientierung der Tiefenkamera relativ zum 3D Modell und zum anderen aus Messungenauigkeiten der von der Tiefenkamera erfassten Distanzmessungen. Daher werden in den folgenden beiden Kapiteln sowohl verschiedene Ansätze zur Bestimmung der Kamerapose diskutiert (und ein präziser Ansatz der Posenbestimmung für den 3D Soll-Ist Abgleich beschrieben), als auch Verfahren zur Reduktion der Messungenauigkeiten (insbesondere in Form der bereits erwähnten 3D Rekonstruktion).

Präzise Erfassung der Kamerapose

Eine präzise Erfassung der Position und Orientierung der Tiefenkamera ist essentiell für die Genauigkeit des 3D Soll-Ist Abgleichs. Daher werden in diesem Kapitel zuerst verschiedene Ansätze zur Erfassung der Kamerapose diskutiert. Daraufhin wird eine präzise Erfassung der Pose der Tiefenkamera durch die Kombination der Tiefenkamera mit einem Messarm beschrieben.

Diskussion von Ansätzen zur Erfassung der Kamerapose Drei verschiedene Ansätze zur Erfassung der Kamerapose werden hinsichtlich ihrer Eignung für 3D Differenzerkennung diskutiert: Bildbasierte Schätzung der Kamerapose, Posenbestimmung durch geometrische Registrierung und Erfassung der Pose mit einem Roboterarm bzw. einem manuell beweglichen Messarm.

Bei einer bildbasierten Posenbestimmung werden charakteristische Merkmale in den erfassten 2D Bildern bestimmt und in einer erfassten Sequenz Bild für Bild detektiert. Hierduch können sowohl die

3D Positionen der erfassten Bildpunkte rekonstruiert als auch die Position der Kamera geschätzt werden. Bei einer geometrischen Registrierung werden dagegen die erfassten 3D Messungen der Tiefenbildkamera mit vorherigen Messungen oder mit einem 3D Modell der Umgebung in Übereinstimmung gebracht, indem die Distanzen zwischen den 3D Messungen und dem 3D Modell (oder den vorherigen Messungen) minimiert werden. Sowohl eine bildbasierte Bestimmung der Kamerapose als auch eine geometrische Registrierung haben den Nachteil, dass charakteristische und eindeutige Strukturen und Merkmale vorhanden sein müssen: Bei einer bildbasierten Bestimmung der Kamerapose im 2D Kamerabild, bei einer geometrischen Registrierung in der 3D Struktur der erfassten Szene. Dies ist jedoch häufig nicht der Fall, etwa bei einfarbigen und wenig texturierten Objektoberflächen oder (im Falle einer geometrischen Registrierung) bei planaren Oberflächen.

Eine dritte Möglichkeit zur Bestimmung der Position und Orientierung einer Tiefenkamera ist es, diese fest an einem mechanischen Messarm zu befestigen. Solch ein Messarm besteht aus mehreren starren Gelenken, die durch Rotationsgelenke miteinander verbunden sind. Die Winkelstellungen der einzelnen Rotationsgelenke werden mit Hilfe von Winkelgebern erfasst. Hierdurch wird die Position und Orientierung der Messspitze des Arms erfasst. Anhand einer sogenannten Hand-Auge Kalibrierung (welche die relative Transformation zwischen der Messspitze und einer starr daran befestigten Kamera bestimmt) kann anhand der Pose des Messarms auch die Pose der Kamera berechnet werden. Ein Messarm kann entweder durch Programmierung gesteuert werden (dies ist bei Roboterarmen der Fall), oder manuell durch einen Benutzer bewegt werden.

Eine Posenbestimmung mit einem Messarm hat den Vorteil, dass die Pose der Tiefenkamera auch dann robust bestimmt werden kann, wenn keine charakteristischen bildbasierten oder geometrischen Merkmale von der Tiefenkamera erfasst wurden. Bei Verwendung eines Messarms wird keine Rechenleistung für die Bestimmung der Pose benötigt, so dass mehr Rechenkapazitäten für die anderen algorithmischen Komponenten zur Verfügung stehen (die Pose des Arms wird von diesem intern berechnet und direkt ausgegeben). Darüber hinaus ist die Genauigkeit der so erfassten Pose höher als die Genauigkeit einer bildbasierten oder geometrischen Bestimmung der Kamerapose: Während Posen, die mit bildbasierten Ansätzen oder anhand von Tiefenbildern geschätzt wurden, um mehrere Millimeter bis Zentimeter von der tatsächlichen Position abweichen können, garantiert ein Faro Platinum Messarm beispielsweise eine Genauigkeit von 0.1mm.

Ein Messarm kann daher für einen präzisen 3D Soll-Ist Abgleich genutzt werden. Aus diesem Grund wird im Folgenden näher auf die Posenbestimmung einer Tiefenkamera anhand eines Messarms eingegangen.

Kombination einer Tiefenkamera und eines Messarms: Hand-Auge Kalibrierung Da ein Messarm die Pose der Messspitze des Arms mit einer hohen Präzision bestimmen kann, besteht die wesentliche Fehlerquelle bei Verwendung solch eines Messarms in der Präzision der Hand-Auge Kalibrierung zwischen der Messspitze und der an der Messspitze befestigten Tiefenkamera. Tiefenkameras erfassen pro Pixel sowohl einen Distanzwert als auch einen Intensitätswert (welcher der an diesem Pixel gemessenen Helligkeit entspricht). Daher kann die Hand-Auge Transformation zwischen einer Tiefenkamera und einem Messarm entweder anhand der 3D Messwerte oder anhand des Intensitätsbildes berechnet

werden, das von der Tiefenkamera erfasst wurde. Die Bestimmung der Hand-Auge Transformation anhand des Intensitätsbildes entspricht dabei der Berechnung der Hand-Auge Transformation zwischen einer Farbkamera und einem anderen Gerät.

Um eine Aussage darüber treffen zu können, ob eine Hand-Auge Kalibrierung anhand der von der Tiefenkamera erfassten 2D Messdaten eine höhere Präzision liefert als eine Kalibrierung anhand der 3D Messdaten (oder umgekehrt), wird eine vergleichende Evaluierung beider Ansätze benötigt. Daher werden zwei Hand-Auge Kalibrieralgorithmen beschrieben, die auf dem selben Kalibrierprinzip basieren. Der einzige Unterschied zwischen beiden Algorithmen besteht darin, wie diese die Pose der Tiefenkamera bestimmen (entweder anhand des Intensitätsbildes oder anhand der erfassten 3D Messungen). Dadurch sind beide Algorithmen direkt vergleichbar, was eine vergleichende Evaluierung ermöglicht.

Die quantitative Evaluierung zeigt, dass sowohl bildbasierte als auch 3D Daten basierte Algorithmen zur Hand-Auge Kalibrierung zwischen einem Messarm und einer Tiefenkamera akkurate Ergebnisse liefern. Hinsichtlich einer Evaluierung anhand von 3D Daten liefert die 3D Datenbasierte Hand-Auge Kalibrierung präzisere Ergebnisse.

Diese bessere Genauigkeit ist jedoch mit einem deutlich höheren Aufwand für die Hand-Auge Kalibrierung verbunden: Als Voraussetzung für den in dieser Arbeit beschriebenen Ansatz zur geometrischen Hand-Auge Kalibrierung werden sowohl ein dreidimensionaler Kalibrierkörper als auch ein exaktes 3D Modell dieses Kalibrierkörpers benötigt. Darüber hinaus muss die Oberfläche des 3D Modells mit der Messspitze des Messarms abgetastet werden, um Oberflächeninformationen zu erfassen, die für die Registrierung des Kalibrierkörpers und des virtuellen 3D Modell des Kalibrierkörpers benötigt werden. Der auf gemessenen Tiefendaten basierende 3D Ansatz zur Hand-Auge Kalibrierung benötigt daher eine deutlich aufwendigere Vorbereitungsphase als der 2D bildbasierte Ansatz: Dieser setzt lediglich das Anbringen eines 2D Markers auf einer planaren Fläche voraus sowie eine Erfassung der 3D Koordinaten seiner vier Eckpunkte mit der Messspitze des Messarms.

Darüber hinaus ist der auf erfassten Tiefenbildern basierende 3D Ansatz zur Hand-Auge Kalibrierung zwischen einem Messarm und einer Tiefenkamera deutlich rechenaufwendiger als der 2D bildbasierte Ansatz. Mit einer unoptimierten C++ Implementierung benötigt der 3D Ansatz mehr als einen Tag, wenn 500 Tiefenbilder mit einer Auflösung von $640 \cdot 480$ Pixeln zum Zweck der Kalibrierung möglichst genau mit dem 3D Modell des Kalibrierkörpers registriert werden. Mit dem bildbasierten Ansatz kann die Hand-Auge Kalibrierung bei der gleichen Anzahl von 2D Bildern dagegen in wenigen Sekunden bestimmt werden. Der bildbasierte Ansatz ist somit zwar etwas weniger präzise, jedoch deutlich schneller berechnbar und auch deutlich weniger aufwendig hinsichtlich seiner Durchführung.

Reduktion von Messungenauigkeiten

Die Genauigkeit der von Tiefenkameras erfassten Distanzmessungen wird sowohl durch Rauschen als auch durch systematische Messfehler eingeschränkt. Daher werden Ansätze zur Reduktion von Rauschen und systematischen Messfehlern vorgestellt und hinsichtlich ihrer Eignung für einen echtzeitfähigen 3D Soll-Ist Abgleich diskutiert. Darüber hinaus wird die Integration eines 3D Rekonstruktionsverfahrens in den Soll-Ist Abgleich vorgestellt.

Diskussion von Ansätzen zur Reduktion von Messfehlern und Messungenauigkeiten Systematische Messfehler von Tiefenkameras können durch eine Kalibrierung der Tiefendaten reduziert werden. Analog zur intrinsischen Kalibrierung einer Kamera werden hierbei zuerst in einem einmaligen Kalibrierungsschritt Tiefenbilder erfasst und mit Referenzdistanzen verglichen. Die systematischen Abweichungen zwischen den gemessenen und den tatsächlichen Distanzen werden in Form von Parametern, Funktionen oder Tabellen gespeichert. Anschließend können die so erfassten Werte zur Laufzeit genutzt werden, um die von Tiefenkameras gemessenen Distanzwerte zu korrigieren.

Im Gegensatz zu systematischen Messfehlern kann Rauschen in Tiefenbildern anhand von Superresolution oder durch 3D Rekonstruktion reduziert werden. Bei Superresolution wird ein Tiefenbild entweder mit den Farbinformationen eines zusätzlichen, höher aufgelösten 2D Bildes kombiniert oder mit zusätzlichen Tiefenbildern, die aus leicht unterschiedlichen, aber nahe beieinanderliegenden Kamerapositionen aufgenommen wurden. Hierdurch kann die Auflösung und die Genauigkeit der Tiefenwerte erhöht werden. Die Kombination eines Tiefenbildes mit einem höher auflösenden Farbbild setzt jedoch voraus, dass die Distanzwerte mit den Farbwerten korrelieren (etwa, dass Kanten im Farbbild auch dreidimensionalen Kanten entsprechen). Diese Bedingung ist häufig nicht erfüllt, was zu einer Verschlechterung statt einer Verbesserung der Genauigkeit führt. Bei Superresolution durch die Kombination mehrerer Tiefenbilder besteht dieses Problem nicht. Allerdings bestehen hierbei Anforderungen an die Positionen, von denen aus die zu kombinierenden Tiefenbilder aufgenommen werden müssen, die für eine frei bewegte Kameraposition nicht garantiert werden können. Darüber hinaus sind aktuelle Superresolution Algorithmen nicht echtzeitfähig.

Newcombe et al. stellten 2011 einen 3D Rekonstruktionsalgorithmus für Tiefenbilder vor [NIH*11]. Im Gegensatz zu Superresolution ermöglicht dieser eine dichte, echtzeitfähige 3D Rekonstruktion von Oberflächen. Aus diesem Grund wird in dieser Arbeit nicht Superresolution, sondern eine Adaption dieses Rekonstruktionsalgorithmus eingesetzt, um die Daten mehrerer Tiefenbilder zu kombinieren und somit Messungenauigkeiten zu verringern. Im Gegensatz zu der von Newcombe et al. beschriebenen Posenbestimmung mittels geometrischer Registrierung wird die Kamerapose in dieser Arbeit nicht geometrisch, sondern durch die Kombination der Tiefenkamera mit einem Messarm bestimmt.

Reduktion von Messungenauigkeiten durch 3D Rekonstruktion Um die 3D Objektoberflächen der erfassten Szene zu rekonstruieren, wird der erfasste 3D Raum in ein diskretes Voxel Grid aufgeteilt. Jedes Voxel speichert den Wert einer diskretisierten "Truncated Signed Distance Function"(TSDF). Der Wert der TSDF eines Voxels entspricht der Distanz des Voxelzentrums zur nähesten rekonstruierten

Objektoberfläche. Punkte, welche exakt auf der rekonstruierten Objektoberfläche liegen, haben den Wert 0. Für 3D Punkte mit einer Distanz ungleich Null spezifiziert das Vorzeichen, auf welcher Seite der Objektoberfläche sich das Voxelzentrum befindet.

Sobald ein neues Tiefenbild erfasst wurde, wird dieses zuerst mit dem rekonstruierten 3D Modell registriert. In dieser Arbeit wird diese Registrierung anhand der Pose vorgenommen, die durch die Kombination der Tiefenkamera mit dem Messarm bestimmt wurde. Alternativ lässt sich die Pose auch durch optisches Kameratracking oder durch geometrische Registrierung bestimmen. Nach der Registrierung des neuen Tiefenbildes mit der bisherigen 3D Rekonstruktion wird die Rekonstruktion entsprechend der neu erfassten Daten aktualisiert. Hierfür werden die neu erfassten 3D Punkte in das Voxel Grid transformiert, um darauf basierend den TSDF Wert für jedes Voxel neu zu berechnen.

Da es sich bei der TSDF um eine implizite Oberflächenrepräsentation handelt, liegen die Oberflächeninformationen nicht explizit vor. Daher werden diese extrahiert, indem durch Ray Casting ein Tiefenbild aus Sicht der aktuellen Kamerapose berechnet wird. Dieses ersetzt das Tiefenbild, das von der Tiefenkamera gemessen wurde.

Quantitative Evaluierung

Die Genauigkeit des 3D Soll-Ist Abgleichs wird sowohl durch eine Simulation evaluiert als auch durch eine quantitative Evaluierung von Sequenzen, die mit Tiefenkameras aufgenommen wurden.

Simulation Die Genauigkeit des 3D Soll-Ist Abgleiches hängt unter anderem von der Genauigkeit der intrinsischen Kalibrierung, der geschätzten Kamerapose und von Messungenauigkeiten der Tiefenkamera ab. Um den Einfluss dieser Parameter auf die Gesamtgenauigkeit zu ermitteln, wurden Simulationen durchgeführt, welche jeweils einen dieser Parameter variieren. Da die Einflüsse dieser Parameter auf die Gesamtgenauigkeit des Soll-Ist Abgleichs auch von der Geometrie der jeweils erfassten dreidimensionalen Szene abhängen, werden zum einen Simulationsergebnisse für eine planare Fläche (orthogonal zur Blickrichtung der Tiefenkamera) und für eine virtuelle Kameraposition im Zentrum einer Kugel dargestellt. Zusätzlich zu diesen elementaren geometrischen Formen wird anhand einer industriellen Brennstoffzelle exemplarisch gezeigt, wie sich Ungenauigkeiten der einzelnen Parameter bei einem komplexen 3D Objekt auf die Gesamtgenauigkeit auswirken können.

Evaluierung anhand aufgenommener Tiefenbildsequenzen In Ergänzung zu der Simulation, welche den Einfluss verschiedener Parameter auf die Gesamtgenauigkeit des Soll-Ist Abgleichs quantifiziert, wurde durch die Evaluierung aufgenommener Tiefenbildsequenzen ermittelt, welche Genauigkeit mit den vorgeschlagenen Verfahren bei Verwendung aktueller Tiefenkameras erreicht werden kann. Hierfür wurde die Genauigkeit sowohl mit einer "Time-of-Flight"Kamera (SwissRanger4000) evaluiert als auch mit einer Tiefenkamera, welche die Distanz mit strukturiertem Licht erfasst (Kinect).

Tabelle 0.2 zeigt die Genauigkeit des Soll-Ist Abgleichs, die sich bei Verwendung einer Kinect Tiefenkamera ergibt. Während die erste Spalte die jeweilige Distanz der Kamera zur Oberfläche angibt,

stellt die zweite Spalte die Genauigkeit dar, welche mit dem grundlegenden Ansatz zur 3D Differenzerkennung erreicht wird (bildbasierte Bestimmung der Kamerapose mit einem Marker, ohne 3D Rekonstruktion). Die beiden darauf folgenden Spalten geben die Genauigkeit an, die sich ergibt, wenn jeweils eines der beiden vorgeschlagenen Verfahren zur Erhöhung der Genauigkeit des Abgleichs eingesetzt wird (präzise Posenbestimmung durch Kombination der Tiefenkamera mit einem Messarm, beziehungsweise Reduktion von Messungenauigkeiten durch 3D Rekonstruktion).

Die letzte Spalte von Tabelle 0.2 stellt die Genauigkeit dar, welche durch eine Kombination dieser beiden Verfahren erreicht wird. Durch die Kombination der beiden Verfahren zur Verbesserung der Genauigkeit wird der Fehler gegenüber dem grundlegenden Ansatz (markerbasierte Posenbestimmung, ohne 3D Rekonstruktion) halbiert.

Distanz Kinect Kamera zur Oberfläche	**Pose: Marker, ohne 3D Rekonstruktion**	**Pose: Marker, mit 3D Rekonstruktion**	**Pose: Messarm, ohne 3D Rekonstruktion**	**Pose: Messarm, mit 3D Rekonstruktion**
450-599	6.54	7.76	3.70	1.96
600-749	10.34	10.71	4.88	4.41
750-899	8.40	6.50	6.87	4.80
900-1049	11.34	8.54	10.84	7.30
1050-1199	23.39	13.37	18.97	11.88
1200-1349	38.56	22.81	26.24	14.31
1350-1499	48.78	39.85	38.26	20.31
1500-1649	64.49	48.35	50.58	24.11

Tabelle 0.2.: Abweichung (Median) zwischen 3D Messwerten und dem wahren Abstand zwischen der Tiefenkamera und der Objektoberfläche. Alle Werte sind in Millimetern angegeben.

Neben der Verbesserung der Genauigkeit hat die Integration der beiden Verfahren den Vorteil, dass die Bestimmung der Kamerapose nicht aufgrund einer ungünstigen Struktur der erfassten Szene (etwa wenig charakteristische Merkmale, homogene Strukturen) fehlschlagen kann. Darüber hinaus enthält die dargestellt Differenzvisualisierung weniger Lücken: Regionen, an denen die Tiefenkamera im aktuellen Bild keine Distanzwerte erfassen konnte, werden durch die 3D Rekonstruktion ersetzt.

Fazit

In dieser Arbeit wurde ein echtzeitfähiges Verfahren für einen tiefenbildbasierten 3D Soll-Ist Abgleich vorgestellt. Hierbei handelt es sich um das erste Verfahren, welches einen dichten Echtzeit 3D Abgleich nicht nur für statische Betrachungspositionen, sondern auch für eine vom Benutzer bewegte Tiefenkamera ermöglicht. Frühere Verfahren waren entweder auf statische Betrachungspositionen beschränkt (so dass jede neue Betrachungsposition manuell aufwendig neu eingemessen werden musste), nicht echtzeitfähig oder umfassten nur eine rein visuelle Überlagerung von 2D Bildern mit einem 3D Modell, ohne die Erfassung von 3D Messdaten.

Das in dieser Arbeit vorgestellte Verfahren ermöglicht einen echtzeitfähigen 3D Soll-Ist Abgleich durch die Kombination von Computer Vision und Computer Graphik. Hierfür wird die Position und Orientierung der Kamera relativ zum 3D Modell in Echtzeit erfasst. Anschließend ermöglicht ein Computer Graphik basiertes Analyse-durch-Synthese Verfahren eine effiziente und echtzeitfähige Zuordnung aller gemessenen Distanzwerte zu entsprechenden 3D Punkten auf dem 3D Modell. Durch die Kombination der Posenschätzung und des Analyse-durch-Synthese Verfahrens können Tiefenbilder mit 600.000 Tiefenmessungen in weniger als 15 Millisekunden mit einem komplexen 3D Modell verglichen werden, das 2,5 Millionen Dreiecke umfasst.

Um nicht nur einen echtzeitfähigen, sondern auch einen präzisen 3D Soll-Ist Abgleich zu ermöglichen, wurden in dieser Arbeit darüber hinaus Ergänzungen des Soll-Ist Abgleichs vorgestellt, welche einen 3D Abgleich mit hoher Genauigkeit ermöglichen.

Die beiden wesentlichen Aspekte, welche die Präzision des 3D Soll-Ist Abgleiches einschränken, sind zum einen Ungenauigkeiten der berechneten Pose der Tiefenkamera relativ zum 3D Modell und zum anderen Rauschen und systematische Messfehler in den von der Tiefenkamera erfassten Tiefenbildern. Aus diesem Grund wurden in dieser Arbeit Verfahren vorgeschlagen und evaluiert, welche sowohl eine genaue Bestimmung der Pose einer Tiefenkamera ermöglichen als auch die Ungenauigkeiten der von Tiefenkameras erfassten Tiefendaten verringern.

Für eine möglichst präzise Bestimmung der Position und Orientierung der Tiefenkamera wurden bildbasierte Posenbestimmung, geometrische Registrierung und Posenbestimmung durch eine Koordinatenmessmaschine (bzw. einen Messarm) diskutiert und Verfahren zur Posenbestimmung durch Kombination einer Tiefenkamera mit einer Koordinatenmessmaschine vorgeschlagen sowie vergleichend evaluiert. Zur Verringerung von Messungenauigkeiten wurde der 3D Soll-Ist Abgleich durch eine 3D Rekonstruktion ergänzt, welche durch eine massive Parallelisierung auf der GPU die erfasste Szene in Echtzeit (während des Soll-Ist Abgleiches) rekonstruiert. Hierdurch werden nicht nur Rauschen und systematische Messfehler verringert, sondern auch Bereiche des Tiefenbildes ergänzt, an denen im aktuellen Bild keine Erfassung von Tiefendaten möglich war.

Die Genauigkeit des 3D Soll-Ist Vergleiches wurde quantitativ evaluiert. Hierfür wurde zum einen eine Simulation durchgeführt, um den Einfluss einzelner Faktoren (u.a. intrinsische Parameter, Ungenauigkeiten in der Bestimmung der Kamerapose oder der Rauschen in den erfassten Messungen) auf die Gesamtgenauigkeit zu quantifizieren. Darüber hinaus wurde die Genauigkeit anhand von aufgenommenen Tiefenbildsequenzen quantitativ evaluiert, sowohl für ein einfaches Setup (bildbasierte Posenschätzung, ohne 3D Rekonstruktion) als auch für die Variante, welche auf eine möglichst hohe Genauigkeit ausgerichtet ist (Posenschätzung anhand einer Kombination der Tiefenkamera mit einem präzisen Messarm, mit 3D Rekonstruktion). Aus einem Meter Messdistanz und bei Verwendung einer auf strukturiertem Licht basierenden Tiefenkamera (Kinect) können mit dem einfachen Setup Abweichungen ab 8 bis 24 Millimetern erkannt werden. Mit den vorgeschlagenen Verfahren zur Genauigkeitsverbesserung können bei Verwendung der gleichen Kamera dagegen bereits Abweichungen ab 4 bis 12 Millimetern erkannt werden.

Contents

1. Introduction

3D difference detection is the task to verify whether the 3D geometry of a real object exactly corresponds to a virtual 3D model of this object.

Three dimensional difference detection can be used both for evaluating the virtual 3D model and the real object. If the virtual 3D model specifies the shape that a real object should have, 3D difference detection can be used to check the accuracy of the real object. On the other hand, 3D difference detection can also be used to evaluate the accuracy of the virtual 3D model.

Detecting differences between a real object and a 3D model of this object is important in a wide range of application areas such as architecture and construction, industrial applications and 3D modeling. Examples for applications in which the real object needs to be checked are:

- **Assembly control:** After a worker has assembled several parts of an object, geometric difference detection between a reference 3D model and the assembled object can be used to check if each component was attached at the correct position. The same approach can also be used to immediately detect differences during the assembly process itself. Such a discrepancy check can for example be used to detect if a tube or a pipe was attached to a different position than intended.
- **Manufacturing:** Given a 3D model of the manufactured object, a 3D discrepancy check can detect differences between the 3D model and the constructed object which might occur due to inaccuracies in the manufacturing process.
- **3D difference detection for construction:** After a building element or a technical installation was constructed, 3D difference detection can be used to check whether the constructed and installed elements really comply to the 3D specification.

On the other hand, in other applications not the real object but the 3D model needs to be checked.

- **Prototyping**: In prototyping processes, sometimes physical prototypes are created to conduct certain evaluations for which a virtual simulation is not sufficient. As part of these processes, the physical prototypes can be changed. In this case, the 3D model needs to be updated according to the changes of the physical prototype. Here, 3D difference detection can be used to check where the 3D shape of the altered prototype differs from the shape of the 3D model. This helps to detect parts of the 3D model where the 3D model needs to be updated.
- **3D modeling:** The process of creating a virtual 3D model is called 3D modeling. There are applications such as Augmented Reality applications for which it is very useful to have an accurate 3D model of a real scene. Augmented Reality applications augment 2D camera images in real time with additional information. A worker repairing a machine can for example point a video

camera towards the machine to get a 3D visualization of the next repair step augmented onto the current 2D camera image. A 3D model of the real scene is useful for two different aspects. First, it can be used to estimate the position and orientation of the camera relative to the captured scene (which is required to align the augmentation with the real image). Second, a 3D model of the real scene can be used to render the augmentation in a more realistic way by taking into account shadows or occlusions [FKOJ11] [Kah13].

3D difference detection is useful for 3D modeling because it can be used to check if the shape of the created 3D model exactly matches the real scene. Such a 3D difference detection step can not only be applied after the creation of the 3D model was completed, but can also be applied as part of the 3D modeling process [Kah13].

In current state of the art approaches (for example in industrial applications), 3D difference detection often is an offline task. Due to their high measurement precision, high-end laser scanners are the technology of choice for offline 3D difference detection. State of the art laser scanners only capture depth measurements along a single scan line. To acquire a dense 3D point cloud from a single point of view, these point- or line based scanners need to sequentially scan the environment, either by automatically rotating parts of the scan head [Far13a] [Lei13] or with hand-held approaches [Far13b]. Each scan takes several seconds to several minutes.

1.1. Real-time 3D difference detection

This thesis introduces real-time 3D difference detection based on depth images. It describes how 3D differences can be detected on-the-fly. In contrast to offline approaches, real-time 3D difference detection provides an immediate feedback whether the 3D object matches the 3D model or not. With the proposed concept, 3D differences can be detected on-site and from arbitrary viewing positions. Furthermore, the viewing positions can be changed dynamically during the 3D difference detection process. In contrast to offline approaches, the user is not restricted to a single predefined viewpoint, but can arbitrarily change the viewpoint during the 3D difference detection to inspect details or to detect differences at different parts of the inspected object.

The 3D difference detection concept described in this thesis introduces 3D difference detection with 3D depth cameras. 3D imaging with depth cameras is a 3D measurement technology which has been subject to significant technological progress in the last years. In contrast to laser scanners, depth cameras acquire dense 3D point clouds at interactive update rates of up to 30 frames per second. State of the art depth cameras are either based on the time-of-flight principle [OLB06] [KBKL09] or use structured light to estimate the depth, such as the Kinect depth camera [ZSMG07].

Whereas depth cameras are commonly used in the consumer mass market (for example in gaming applications), up to now they are only seldomly used in industrial applications. This is mainly due to the poor measurement quality of these depth cameras. Depending on the surface properties of the measured object, the measured distance can differ from the real distance by several centimeters at a distance from 0.5 to 5.0 meters.

Despite their limited measurement accuracy, the application of depth cameras for 3D difference detection seems to be promising. Depth cameras are low-cost devices which are able to capture the 3D surface of an object in real time, which can be moved during the 3D data acquisition and which are eye-safe for the users. In contrast to stationary laser scanners, the user can move the depth camera around the object. Thus, if suitable approaches for applying this technology are available, the user is not restricted to an offline approach any more. Such an offline approach requires that the object is scanned first. Then, differences can be detected in a second step, at a later point of time. With depth camera based real-time 3D difference detection, the user can inspect differences at arbitrary parts of the object in real time. Furthermore, the user does not need to wait until the 3D scan has finished before changing the viewpoint of the 3D difference detection and before scanning a different part of the inspected object.

Depth cameras provide the technological basis for real-time 3D difference detection in terms of 3D data acquisition devices. However, the availability of such 3D scanning devices is necessary, but not sufficient for real-time 3D difference detection. Previous approaches cannot be applied directly for real-time 3D difference detection based on depth images. They either require manual user input for each changed 3D data acquisition position [Bos08] [Bos10] or leave the 3D difference detection task up to a human [GBSN09] [FG11], without actually measuring the 3D surface of the inspected object. Therefore, to use this 3D measurement technology for real-time 3D difference detection, new approaches need to be researched and evaluated. The lack of existing approaches for depth image based, real-time 3D difference detection opens up a new field of research, which is approached by this work.

This thesis introduces a general concept for real-time 3D difference detection with a depth camera. Due to the limited measurement accuracy of depth cameras, improving the overall difference detection accuracy is a major challenge for real-time difference detection. Therefore, different approaches for enhancing the 3D difference detection accuracy are proposed and their integration in the 3D difference detection concept is described. Furthermore, this thesis provides a quantitative evaluation of the 3D difference detection accuracy for different setups of the 3D difference detection concept.

Benefits

The main benefits of the depth image based 3D difference detection as presented in this thesis are:

- In contrast to previous approaches, 3D differences can be detected in **real time**.
- The user is not restricted to a single viewpoint, but can inspect differences at arbitrary parts of the object from **arbitrary viewpoints** in real time. In contrast to previous approaches (such as difference detection with stationary laser scanners), the user can move the depth camera around the object or move the camera closer to the object to have a look at relevant details or to bypass visual occlusions.
- While depth cameras have a lower measurement accuracy than high end laser scanners, they are also much **cheaper**. A laser scanner costs about 100 times as much as a depth camera.

In the remainder of this chapter, first a problem definition of 3D difference detection is provided. Then, the main research questions of this thesis are defined. Finally, an outline of the chapters of this thesis is given.

1.2. Problem definition

The 3D difference detection problem is defined as follows.

Given:

- A **real object**.
- A **virtual 3D model** of this object in a well-defined state.

Find:

Dense 3D differences between the shape of the real object and the shape of the 3D model.

In this definition, the word "real object" denotes a tangible object, which exists physically. In contrast, the "3D model" does not denote a physical object. It is a pure virtual representation of the physical object, stored in computer memory.

In view of possible configurations which might alter the shape of the 3D model, we assume that the 3D model is in a well-defined state: for each point in time, the configuration of the 3D model at this point in time is known. This is obviously true for rigid 3D models and for 3D models which do not alter their shape during the 3D difference detection process. All approaches for 3D difference detection described in this thesis can be applied to these 3D models. Furthermore, all approaches except the 3D surface reconstruction for precision enhancement (Section 5.2) can also be applied to 3D models which alter their shape during the 3D difference detection (as long as the shape of the 3D model is well-defined for each point in time).

The word "dense" distinguishes the difference detection from sparse 3D difference detection. If 3D differences are only detected at a few single 3D points (as with sparse 3D difference detection), there is a lack of information about differences at the shapes between these single 3D points. In contrast, dense 3D difference detection provides continuous and complete difference information.

In this thesis, in addition to the provided definition, the 3D difference detection is furthermore subject to the following requirements:

- The 3D difference detection should be real-time capable (the 3D differences detection should be updated several times per second).
- The representation of the 3D model should not be restricted to a certain format (such as a triangle mesh representation). Instead, the 3D difference detection should be applicable for 3D models in arbitrary formats.
- The 3D shape differences should be detectable both for convex and for concave parts of the object.

1.3. Research questions

The three main research questions addressed by this thesis are:

Q1 ***How can 3D differences be detected in real time and from arbitrary viewpoints using a single depth camera?***

Q2 ***Extending the first question, how can 3D differences be detected with a high precision?***

Q3 ***Which accuracy can be achieved with concrete setups of the proposed concept for real time, depth image based 3D difference detection?***

This thesis approaches the three main research questions by dividing them into several subquestions.

Concept for real-time 3D difference detection

To answer the first research question (Q1), two main contributions are required. First, a solution is required for mapping 3D measurements (acquired by a moving depth camera) onto an arbitrary 3D model in real time. Second, such a 3D mapping algorithm needs to be complemented by other algorithmic components (such as real-time pose estimation and 3D difference calculation). To address both aspects of the first research question, Q1 is divided into two subquestions:

Q1.1 How can the 3D measurements of a depth camera be mapped onto an arbitrary 3D model in real time?

Q1.2 Given a mapping of 3D measurements onto a 3D model, how can 3D differences be detected in real time for a moving depth camera?

Enhancing Precision

To approach the second main research question (Q2), the factors which limit the accuracy of 3D difference detection need to be taken into account. The accuracy of 3D difference detection is limited by pose estimation inaccuracies and measurement inaccuracies. Therefore, both pose estimation inaccuracies and measurement inaccuracies should be reduced as far as possible to ensure precise 3D difference detection. For this reason, the next research questions refer to the algorithmic adaptation and integration of approaches which enhance the accuracy of depth image based 3D difference detection. They address both the accuracy of the camera pose estimation and an algorithmic enhancement of the 3D measurements.

Q2.1 How can precise pose estimation be integrated in the 3D difference detection?

Q2.2 How can measurement inaccuracies be reduced in the context of 3D difference detection?

Quantitative Evaluation

The third research question (Q3) addresses the accuracy of the proposed 3D difference detection. On the one hand, the overall accuracy is influenced by the accuracy of estimated parameters, such as the intrinsic and extrinsic parameters of the depth camera. On the other hand, the overall accuracy depends on the measurement accuracy of depth cameras. In order to investigate these aspects, Q3 is divided into two subquestions. While the first subquestion addresses a theoretical accuracy analysis of these factors, the second subquestion addresses the accuracy which can be achieved with actual state of the art depth cameras.

Q3.1 How do pose estimation inaccuracies and 3D measurement inaccuracies influence the overall achievable accuracy of depth image based 3D difference detection?

Q3.2 Which accuracy can be achieved with different setups of the 3D difference detection, using real sequences captured with state of the art depth cameras?

1.4. Thesis outline and main contributions

Chapter 2 provides an overview of the state of the art in 3D data acquisition and difference detection, as well as an overview of computer vision concepts which are used in this thesis. The focus of the state of the art in 3D data acquisition is on depth cameras, which capture dense 3D point clouds in real time. State of the art depth cameras are either based on the time-of-flight principle or use structured light to estimate the distances. However, the dense, real-time 3D difference detection introduced in this thesis is not restricted to specific kinds of depth cameras. It can be applied with any kinds of depth cameras, irrespective of their internal mode of operation. The second part of Chapter 2 describes the state of the art in 2D and 3D difference detection. Previous approaches for difference detection were either restricted to a static camera position, not real-time capable or did not detect dense 3D differences.

Chapter 3 introduces the concept for real-time, dense 3D difference detection. It first addresses the research question Q1.1 by proposing a general real-time 3D difference detection approach. This approach maps the 3D measurements of a depth camera onto an arbitrary 3D model in real time by fusing computer vision (depth imaging and pose estimation) with a computer graphics based analysis-by-synthesis approach. It can be used for any 3D model which can be rendered, independent of the format and the internal representation of the 3D model. To address the research question Q1.2, this chapter introduces the main algorithmic components as well as several concrete instantiations of the proposed, general 3D difference detection concept. Furthermore, this chapter sketches how the proposed 3D difference detection can reduce the gap between 3D modeling and augmented reality. Finally, this chapter describes the main factors which influence the accuracy of depth image based 3D difference detection.

The main contributions of Chapter 3 are:

- A general concept for depth image based, real-time 3D difference detection

- A real-time 3D mapping approach which registers the 3D measurements of a moving depth camera with an arbitrary 3D model in real time.
- Concrete instantiations of the general 3D difference detection concept.
- A description of the factors which influence the accuracy of depth image based 3D difference detection.

Chapter 4 studies the research question Q2.1, which refers to a precise estimation of the position and orientation of the depth camera. This chapter first discusses the suitability of different approaches for pose estimation in the specific context of 3D difference detection. Based on this discussion, the combination of a portable coordinate measuring machine with a depth camera is proposed for 3D difference detection with high precision pose estimation. A prerequisite for combining a coordinate measuring machine (such as a measurement arm) with a depth camera is that the relative transformation between the tip of the measurement arm and the depth camera needs to be known. While state of the art approaches for 2D cameras use the 2D camera image to estimate this transformation, the depth values captured by depth cameras can also be used for this estimation. Therefore, this chapter proposes both a 2D and a 3D data based transformation estimation algorithm and compares both approaches with a quantitative, comparative evaluation.

The main contributions of Chapter 4 are:

- A discussion of different approaches for accurate pose estimation (image based, geometric or with coordinate measuring machines).
- An integration of precise pose estimation in the 3D difference detection concept. The pose estimation is based on the combination of a depth camera with a portable coordinate measuring machine (a measurement arm).
- Both 2D and 3D data based algorithms for estimating the relative transformation between a depth camera and a portable coordinate measuring machine.
- A comparative, quantitative evaluation of the proposed 2D and 3D transformation estimation algorithms.

Chapter 5 addresses the research question Q2.2, which refers to an accuracy improvement of the 3D values measured by the depth camera. First, potential approaches for reducing depth measurement inaccuracies are discussed. This covers approaches which improve the accuracy of measured 3D data for specific depth cameras and approaches which are more generic, by being independent of a specific depth measurement technology. The latter covers both superresolution and 3D surface reconstruction algorithms. Based on this theoretical analysis, a 3D reconstruction algorithm [NIH*11] [IKH*11] is selected. Then, this algorithm is adapted for the specific requirements of 3D difference detection. This on-the-fly 3D reconstruction of the captured object surface reduces measurement inaccuracies by fusing depth measurements from several depth images into a reconstructed object surface estimation.

The main contributions of Chapter 5 are:

- A discussion of potential algorithms for enhancing the accuracy of 3D measurements (3D calibration, depth image filters, superresolution and 3D surface reconstruction).
- An adaptation of a 3D surface reconstruction algorithm for precise 3D difference detection.
- The integration of the adapted 3D reconstruction algorithm in the 3D difference detection process.

Chapter 6 provides a quantitative evaluation of different setups of the proposed 3D difference detection. First, the question Q3.1 is addressed by a simulation which quantifies the effects of different inaccuracies on the overall 3D difference detection accuracy. The effects of inaccuracies in the estimation of the extrinsic camera parameters (rotation and translation) are simulated as well as inaccuracies in the intrinsic camera paramters. Furthermore, the simulation covers the effect of random measurement noise in the depth measurements and the approximation of curved shapes by planar surface representations.

This simulation is complemented by the evaluation of real depth image sequences, captured with state of the art depth cameras. The ground truth based, quantitative evaluation of the real sequences provides answers to Q3.2 by quantifying the 3D difference detection accuracy for different state of the art depth cameras, both for the basic setup and for the precise setup that was proposed in the Chapters 4 and 5.

The main contributions of Chapter 6 are:

- A simulation which quantifies the influence of: the accuracy of the internal camera parameters, pose estimation inaccuracies, measurement inaccuracies and inaccuracies of the 3D model.
- A quantitative evaluation of the proposed 3D difference detection concept using real sequences, both for the basic setup and for the proposed setup that focuses on precision.

Finally, Chapter 7 concludes this thesis with a summary of the main contributions and with an outlook which describes how in future work, the approaches described in this thesis could be extended. For example, the approaches introduced by this work could also be used to find a configuration of a parametrizable 3D models such that the parametrized 3D model matches a real object as well as possible.

2. Background

This chapter first introduces computer vision terms used in the following chapters (Section 2.1). Then, different approaches for measuring the shape of 3D surfaces are described in Section 2.2. For real time 3D difference detection, the most relevant 3D data acquisition techniques are those which capture dense 3D point clouds (respectively depth images) at interactive update rates (several times per second). Thus, real-time 3D data acquisiton devices are described in Section 2.2. In Section 2.3, this chapter provides an overview of the state of the art in 2D and 3D difference detection. Finally, in Section 2.4, this chapter concludes with a discussion about the suitability of state-of-the art approaches for real-time 3D difference detection with a moving depth camera.

2.1. Computer vision

The projection of a 3D scene to a 2D camera image can be approximated with a pinhole camera model (also called "perspective projection"). The pinhole camera model is a mathematical approximation of the imaging process [FP02]. The basic principle of a pinhole camera model is sketched in Figure 2.1. Physically, a pinhole camera can be created by a solid box with a small pinhole on the front plane of the box. Incoming light rays traverse the pinhole and project an inverted image of the captured 3D object onto the back plane of the pinhole camera. Mirroring the back plane at the pinhole provides a virtual image plane, which has exactly the same distance to the pinhole as the back plane. Thus, the image projected onto the virtual image plane is the mirrored image of the back plane.

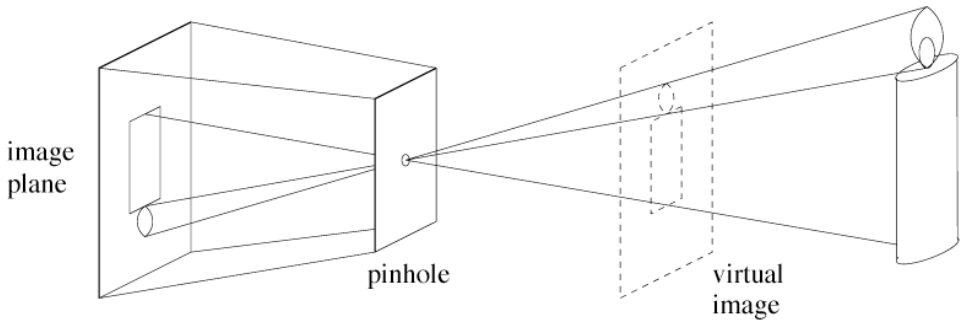


Figure 2.1.: Pinhole camera [FP02].

2.1.1. Perspective camera model

The local coordinate system of a camera is called camera coordinate system (CCS). The origin of the camera coordinate system is located at the optical center of the camera, which is also called the camera's center of projection. The optical center corresponds to the pinhole of the pinhole camera model sketched in Figure 2.1. Per definition, the z-axis of the camera coordinate system is orthogonal to the virtual image plane in front of the camera and intersects the image plane at the principal point (p_x, p_y).

A 3D point observed by the camera gets projected onto the virtual image plane of the camera along a view ray that passes through the 3D point and through the camera's center of projection. Thus, the projection of a 3D point onto the image plane is the intersection of the view ray with the image plane. Mathematically, the projection of a 3D point from the world coordinate system to the virtual image plane is modeled by the extrinsic and intrinsic parameters of a camera.

Extrinsic parameters The extrinsic parameters (R,t) describe a rigid transformation between the camera coordinate system and the fixed world coordinate system. The extrinsic parameters are composed of a rotation R and a translation t, which describe the position and orientation of the camera in the world coordinate system. These parameters change when the camera is moved. The camera pose has six degrees of freedom (three degrees of freedom for the rotation and three degrees of freedom for the translation). The estimation of the extrinsic parameters is called "camera tracking", respectively "camera pose estimation".

The extrinsic parameters (R,t) are defined as the rotation and the translation which map a 3D point M_{WCS} from the world coordinate system to the 3D point M_{CCS} in the camera coordinate system:

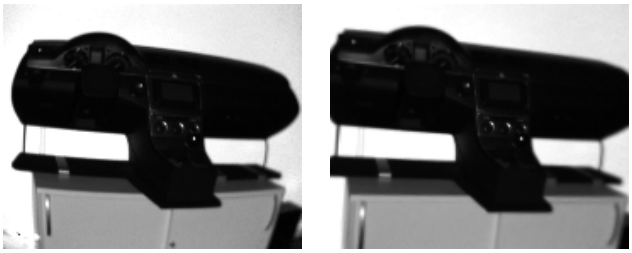
$$M_{CCS} = R \cdot M_{WCS} + t. \tag{2.1}$$

Due to this definition, the extrinsic parameters are not equivalent to the position and rotation of the camera relative to the world coordinate system. However, these values can be easily calculated from the extrinsic parameters. The position of the camera (more exactly, its optical center) is positioned at the origin of the camera coordinate system. Thus, the position of the camera in the world coordinate system (CamPosWorld) and the rotation of the camera relative to the world coordinate system can be calculated from Equation 2.1. R^{-1} is the rotation from the world coordinate system to the camera coordinate system. The position of the camera in the WCS is:

$$CamPosWorld = -R^{-1} \cdot t. \tag{2.2}$$

Intrinsic parameters The intrinsic parameters model the projection of a 3D point M from the camera coordinate system to a projection m in the image coordinate system $(\vec{i}, \vec{j})$ of the camera. With the pinhole camera model, the intrinsic can be represented by five different parameters: by the focal length

(a) Distorted. (b) Undistorted.



(c) Distorted. (d) Undistorted.

Figure 2.2.: Distorted and undistorted intensity images, captured with a SwissRanger 4000 camera.

(f_i, f_j) in both image dimensions, by the principal point (c_i, c_j) and by the skew s. If $\vec{i}$ and $\vec{j}$ are perpendicular, the skew s is is zero. As $\vec{i}$ and $\vec{j}$ are perpendicular in most state-of-the art cameras, the skew s is usually zero. The focal length is the distance between the optical camera center and the virtual image plane, along the principal axis of the camera. The intrinsic camera parameters compose the camera calibration matrix K:

$$K = \begin{pmatrix} f_i & s & c_i \\ 0 & f_j & c_j \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.3}$$

Image undistortion The pinhole model does not take into account radial and tangential distortions caused by camera lenses. Therefore, the projection of a 3D scene onto a 2D camera image is mathematically modeled by a pinhole camera model, extended with additional image undistortion [HS97]. The radial and tangential distortion are modeled with five polynomial distortion coefficients $\kappa_1, ..., \kappa_5$. These distortion coefficients are estimated as part of the intrinsic calibration procedure. They provide a mapping between the distorted and the undistorted image. Therefore, an image can be undistorted by resampling the distorted image with this mapping. Figure 2.2 shows an intensity image captured with a SwissRanger 4000 depth camera, without and with image undistortion.

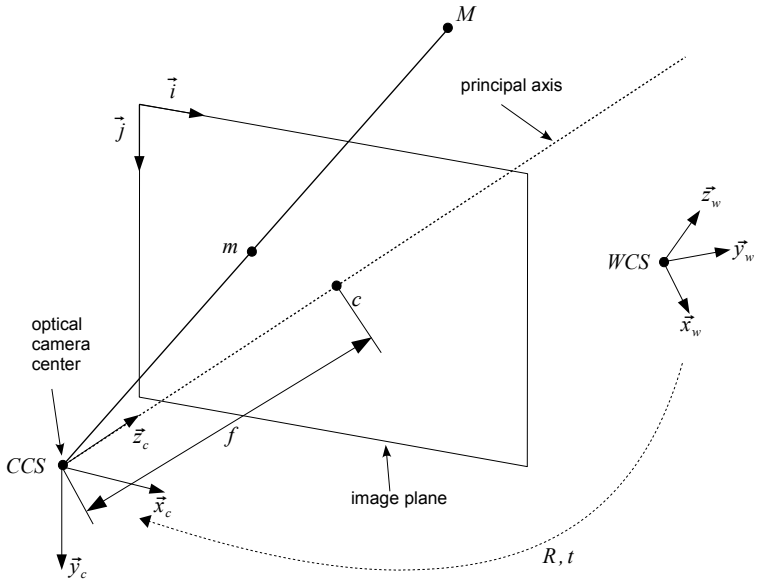


Figure 2.3.: Projection of a 3D point M to the 2D point m on the image plane (adapted from [Wue08]).

Projection from the world to the image coordinate system The projection of a 3D point M (specified in the world coordinate system) to a 2D point m in the image coordinate system $(\vec{u},\vec{v})$ is illustrated in Figure 2.3. First, with the extrinsic data (R,t), the 3D point is transformed from the world coordinate system to a 3D point in the camera coordinate system. Then, the the intrinsic camera calibration parameters $f = (f_x, f_y)$ and $c = (c_x, c_y)$ are used to project the 3D point onto the 2D image coordinate system. In the illustration, the skew s is omitted because $s = 0$.

If $\tilde{M} = (X,Y,Z,1)^T$ and $\tilde{m} = (x,y,1)^T$ denote the homogeneous coordinates of M and m, the projection of M to m can be calculated with the following equation:

$$s_{sc}\tilde{m} = K \cdot \left[R|t\right] \cdot \tilde{M}, \tag{2.4}$$

Here, s_{sc} is a constant scale factor, K is the camera calibration matrix storing the intrinsic parameters and (R,t) are the extrinsic camera parameters. The intrinsic and extrinsic parameters can be combined into a single matrix which is called the "projection matrix" P:

$$P = K \cdot \left[R|t\right], \tag{2.5}$$

2.1.2. Relative transformation between camera poses

The relative transformation between two camera poses (R_1, t_1) and (R_2, t_2) can be derived from the projection of a 3D point from the world coordinate system to the camera coordinate system [Wue08]. Therefore, Equation 2.1 is first formulated for both cameras:

$$\begin{aligned} M_{CCS1} &= R_1 \cdot M_{WCS} + t_1 \Rightarrow M_{WCS} = R_1^{-1} \cdot (M_{CCS1} - t_1). \\ M_{CCS2} &= R_2 \cdot M_{WCS} + t_2. \end{aligned} \tag{2.6}$$

In a next step, M_{WCS} is substituted in the second equation:

$$\begin{aligned} M_{CCS2} &= R_2 \cdot R_1^{-1} \cdot (M_{CCS1} - t_1) + t_2 \\ &= (R_2 \cdot R_1^{-1}) \cdot M_{CCS1} + (t_2 - R_2 \cdot R_1^{-1} \cdot t_1). \end{aligned} \tag{2.7}$$

Thus, the relative transformation between two camera poses represented by extrinsic data can be expressed by $(\Delta R, \Delta t)$:

$$\begin{aligned} \Delta R &= R_2 \cdot R_1^{-1} \\ \Delta t &= t_2 - R_2 \cdot R_1^{-1} \cdot t_1 \end{aligned} \tag{2.8}$$

Vice versa, given the relative transformation $(\Delta R, \Delta t)$, the pose (R_2, t_2) of a second camera can be calculated from the pose (R_1, t_1) of the first camera:

$$\begin{aligned} R_2 &= R_1 \cdot \Delta R \\ t_2 &= R_2 \cdot t_1 + \Delta t \end{aligned} \tag{2.9}$$

2.1.3. Depth images and 3D point clouds

Depth images store a depth value per pixel. There are two different ways to encode depth in a depth image:

1. The depth d stored by a pixel of the depth image represents the euclidean distance between the optical center of the depth camera and the 3D point p_{CCS} in the camera coordinate system. Thus, the stored depth is the length of the view ray from the optical camera center to the 3D point p_{CCS}. This representation corresponds to a perspective projection.
2. The depth value d' is the orthogonal distance between the 3D point p_{CCS} in the camera coordinate system and the optical camera center. As illustrated in Figure 2.3, the z-axis of the camera coordinate system is orthogonal to the image plane. Thus, the stored depth value d' is the z-value of p_{CCS}. This representation corresponds to an orthogonal projection.

Optionally, in addition to the depth values, an intensity (grey) or a color value can be provided for each pixel. For example, time-of-flight depth cameras capture a depth image that provides a depth value and an intensity value for each pixel.

Conversion from depth values to a 3D point cloud Depth images can be converted to 3D point clouds (and vice versa). If the stored depth values represent the euclidean distance between the optical camera center and the 3D point p_{CCS} in the camera coordinate system, the depth d of a pixel m can be converted to p_{CCS} as follows. First, the view ray r through the 2D pixel m is calculated:

$$r = K^{-1} \cdot m. \tag{2.10}$$

Then, the 3D point p_{CCS} is calculated by multiplying the depth value with the normalized view ray:

$$p_{CCS} = d \cdot \frac{r}{\|r\|} \tag{2.11}$$

If the stored depth values represent the z-values of a 3D point p_{CCS} in the depth camera's coordinate system (orthogonal projection), the x and y values of p_{CCS} can be restored as follows. Equation (2.12) transforms the depth value d' of a pixel m in the 2D image coordinate system of the depth camera to a 3D point p_{CCS} in the camera coordinate system. The horizontal respectively vertical focal length is denoted by (f_i, f_j) and the principal point by (c_i, c_j).

$$p_{CCS} = \begin{pmatrix} (p_i - c_i) \cdot \frac{1}{f_i} \cdot d' \\ (p_j - c_j) \cdot \frac{1}{f_j} \cdot d' \\ d' \end{pmatrix} \tag{2.12}$$

Local neighbourhood of depth measurements In contrast to unordered 3D point cloud, depth images provide ordered 3D points with neighbourhood information. Each 3D measurement belongs to

a 2D pixel m in the depth image. Thus, close 3D measurements can be found by inspecting the 3D measurements that belong to the pixels in the local neighbourhood of the 2D pixel m.

2.1.4. Registration / alignment

Registration is the task to transform data from different coordinate systems into a common coordinate system. It is also called alignment [DWJM98] [Fit03] [RC11]. In this thesis, the terms registration and alignment are used interchangeably.

Different kinds of 2D and of 3D data can be registered. For example, a 3D model can be aligned with a 2D image, or several 3D point clouds can be aligned into a common coordinate system. Fitzgibbon pointed out that the major challenge for finding such a transformation is that correspondences between the different data sets are not known a priori [Fit03]. Therefore, a common way to approach the registration problem is the manual selection of corresponding point pairs. Then, these point pairs are aligned by minimizing the distances between the pairs of corresponding points [Ume91]. Instead of points, other features (such as planes) can be used for the alignment as well.

Given a coarse registration, the precise alignment of two 3D point clouds can be refined algorithmically with the Iterative Closest Point algorithm (ICP) [BM92] [CSK05]. This algorithm iteratively repeats two steps: First, corresponding point pairs are automatically selected based on the assumption that 3D points from one point cloud correspond to the closest 3D points of the other point cloud. This assumption is only valid if the 3D points are already coarsely aligned. Thus, an initial coarse alignment is a required prerequisite for applying the ICP algorithm. In a next step, the distances between corresponding point pairs are minimized the same way as for manually selected point pairs [Ume91]. These two steps are repeated iteratively until the registration has converged or until a maximal number of iterations was executed.

2.2. Real-time 3D imaging

Detecting 3D differences between a real object and a virtual 3D model requires 3D measurement devices that can capture the 3D surface of the real object. A large number of techniques have been developed for the acquisition of 3D measurements, either with contact based or with contactless approaches. Using contact based approaches, 3D measurements can be acquired with mechanical probes attached to industrial robots or to coordinate measuring machines. Contactless approaches range from structured light based approaches or laser scanning to the reconstruction of 3D data from several 2D images, for example with stereo vision or with structure from motion, respectively with 3D reconstruction by triangulation. A detailed overview of 3D data acquisition is provided by Sansoni et al. [STD09] and by Bi and Wang [BW10]. While these surveys provide a broad overview of 3D data acquisition in general, this section focuses on dense real-time depth imaging with depth cameras, which acquire dense 3D measurements with interactive update rates. Furthermore, selected 3D data acquisition techniques (contact based 3D measuring, laser scanning and image based 3D reconstruction) are introduced. For further information about other 3D data acquisition techniques, see the surveys by Sansoni et al. [STD09] and by Bi and Wang [BW10].

Contact based 3D measuring: robots and coordinate measuring machines Industrial robots and coordinate measuring machines are contact based devices for the acquistion of 3D measurements [Tan92]. Coordinate measuring machines (CMMs) measure 3D coordinates with a mechanical probe which can be moved, either by a program or by a human. If the movement of a CMM is controlled by a program, CMMs are similar to industrial robots, which can acquire contact based 3D measurements as well [Nof99] [SKK*10]. In contrast, mechanical measurement arms are portable coordinate measuring machines which are moved by a user. Such measurement arms are described in more detail in Section 4. Both industrial robots and CMMs provide the position and orientation of the mechanical probe. Thus, for each time instance, a single 3D measurement of the probe's position (and orientation) is provided. In addition to their mechanical probes, coordinate measuring machines and robots can be equipped with additional devices for contactless 3D data acquisition. For example, measurement arms can be combined with 3D laser scanners [Far13b].

Laser scanning Laser scanners provide a very high measurement precision and can measure differences in the millimeter range at several meters distance. However, state of the art laser scanners only capture depth measurements along a single scan line. To acquire a dense 3D point cloud from a single point of view, these point- or line based scanners need to sequentially scan the environment, either by automatically rotating parts of the scan head [Far13a] [Lei13] or with hand-held approaches [Far13b]. Each scan takes several seconds to several minutes.

While 3D imaging based on stereo vision requires a texture on the scanned objects, laser scanners can accurately measure the shape of untextured objects. In contrast to other passive 3D distance estimation approaches such as shape from shadow or shape from shading [PF06], laser scanners do not require

previous knowledge about the surface reflectance properties of the scanned objects or about the lighting of the scene.

Image based 3D data acquisition 3D points can be reconstructed from several 2D images which were acquired from different camera positions. These 2D images can either be acquired simultaneously (when a multi-camera setup is used) or sequentially (by the movement of a single camera to different positions). The 3D reconstruction of 3D positions of characteristic image features detected in several camera images is called triangulation [FP02]. In order to reconstruct 3D points with triangulation, characteristic 2D image features are detected in several camera images taken from non-coincident positions. If the camera poses with which these 2D images were acquired are known, the 3D position of a 2D feature can be reconstructed by estimating the intersection of the view rays from the optical camera centers through the detected 2D feature positions on the virtual image plane [BBS07]. The estimated 3D positions can then be refined by globally minimizing both the reconstructed 3D points and the estimated camera positions with bundle adjustment [WWK11]. Even though dense reconstruction of 3D points is computationally expensive, Newcombe and Davison recently published an approach for real-time dense 3D reconstruction with a single moving camera [ND10].

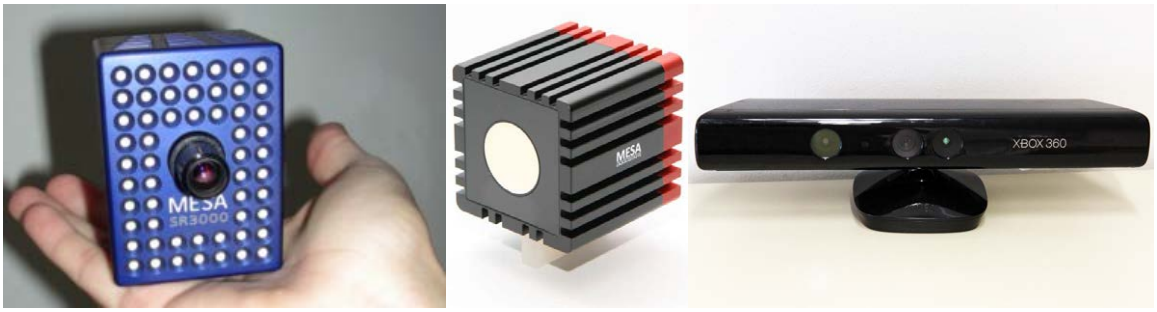
Image based 3D reconstruction has a general major drawback: the reconstruction of a 3D surface is not possible if the surface is not textured enough. The 3D shape of untextured surfaces can not be calculated with triangulation if too few 2D image features can be detected and matched in several 2D images. One possible approach to handle untextured object surfaces is to fill the 3D reconstruction of untextured regions with an interpolation between the parts of the scene which could be reconstructed. This interpolation is based on the assumption that the surface is smooth at the interpolated regions. However, this approach fails if the scene contains too many untextured areas or if the untextured areas are not smooth.

Real-time depth imaging Depth cameras acquire dense 3D measurements in real time. Thus, they provide the technological basis for the real-time 3D difference detection described in this thesis. Just as laser scanners, in order to acquire 3D measurements, depth cameras neither require textures on the scanned surfaces nor previous knowledge about the surface properties. Thus, they can also capture depth information at untextured object surfaces.

The real-time 3D difference detection approach described in this thesis is applicable for any 3D measuring input devices which capture dense depth images in real time. This section describes the two approaches for real-time depth imaging which are currently technologically most advanced and most widely used: both time-of-flight based depth imaging and structured light based depth imaging are described. However, the approaches described in this thesis can also be applied with other, real-time depth imaging devices, irrespective of their internal working principle. The only precondition is that they provide dense depth images in real time.

For example, the depth cameras used for the evaluation of the approaches described in this thesis capture depth images with $176 \cdot 144$ measurements (SwissRanger SR4000 time-of-flight depth camera),

respectively $640 \cdot 480$ measurements (structured light Kinect depth camera). These depth cameras have an update rate of about 10 frames per second (SR4000), respectively 30 frames per second (Kinect).



(a) SR3000. (b) SR4000 [Mes09]. (c) Kinect.

Figure 2.4.: Time-of-flight depth cameras (2.4a, 2.4b) and structured light depth camera (2.4c).

2.2.1. Time-of-flight depth cameras

Time-of-flight cameras emit near-infrared, amplitude modulated light [OLK*04] [OLB06] [HRH08] [KBKL09] [HLCH12]. They capture a depth image as well an intensity image. The intensity depicts the amount of light reflected onto each pixel. In this section, the time-of-flight based depth imaging used by SwissRanger depth cameras is explained [BOG*04] [BS08]. Other time-of-flight depth cameras are based on the same principle, but differ in details (for example in the sampling of the received signal, which is used to calculate the phase shift as explained in the next paragraphs).

The amplitude of the light emitted by time-of-flight cameras is modulated with a continuous-wave modulation. The emitted light gets reflected by the scene and is captured by the image sensor of the camera. Each pixel of the camera's image sensor demodulates the incoming, modulated light field. Thus, the phase delay between the emitted and the returned light signal can be measured. This phase delay is used to calculate complete phase maps. Then, for each pixel, the distance of the optical camera center to the scene is calculated by the phase shift of the reflected light.

Figure 2.5 illustrates the phase measurement principle of time-of-flight cameras. Here, P_A is the sinusoidal signal emitted by the time-of-flight camera. P_B is an amplitude offset which is caused by detected background light that was not emitted by the time-of-flight camera. Due to optical loss (not all reflected lights gets projected back onto the sensor of the depth camera), the light reflected to the sensor of the depth camera has a smaller amplitude than the emitted light. This loss is modeled by the factor k. Thus, the amplitude of the reflected light, which is detected by the sensor, is $k \cdot P_A$. For a time instance t, the emitted signal P_e and the received signal P_r are mathematically modeled by the following equations:

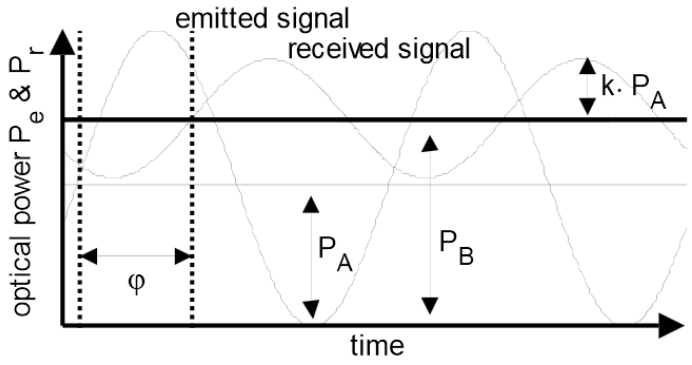


Figure 2.5.: Phase measurement principle of time-of-flight cameras [BOL05].

$$\begin{aligned} P_e &= P_A \cdot [1 + \cos(2\pi f t)] \\ P_r &= P_B + k \cdot P_A \cdot [1 + \cos(2\pi f (t - \tau))]. \end{aligned} \tag{2.13}$$

In this notation, f is the modulation frequency and τ is the round-trip time (the time it takes until the emitted light is reprojected onto the sensor) [BS08]. The round-trip time τ (and thus the distance to the measured object surface) is directly proportional to the phase delay φ between the emitted and the reflected sinusoids:

$$\tau = \frac{\varphi}{2\pi f} \tag{2.14}$$

Given the round-trip time τ, the distance d to the object surface can be calculated with the following equation (c is the speed of light):

$$d = \frac{c \cdot \tau}{2} \tag{2.15}$$

At each pixel, the SwissRanger time-of-flight camera measures the reflected sinusoid at equal timing intervals, four times per period [BOG*04]. The four samples are charge carriers, acquired by the integration of a photo-current over time [BS08]. They provide sufficient information to calculate all required parameters of the received signal P_r. Blanc pointed out, that in practice, these four samples are not only acquired for a single period [BOG*04]. Instead, in order to increase the signal to noise ratio and thus the measurement accuracy, they are summed over hundreds or thousands periods.

The amplitude A, the offset P_B and the phase shift φ can be calculated with the following equations if the four measured samples are denoted by A_0, A_1, A_2 and A_3:

$$A = \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2} \tag{2.16}$$

$$P_B = \frac{A_0 + A_1 + A_2 + A_3}{4} \tag{2.17}$$

$$\varphi = \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \tag{2.18}$$

The measured amplitude A corresponds to the amplitude P_A of the emitted signal, diminuished by a factor k ($A = k \cdot P_A$). Thus, the ratio of P_A and A is the "signal to background" ratio, which can be used to estimate the confidence of the measured depth values.

P_B, calculated with Equation 2.17, is the overall amount of light captured by the pixel of the sensor (the reflected light emitted by the time-of-flight camera and the background light). Thus, P_B represents the intensity measured by this pixel of the depth camera [BOG*04]. The intensity image is similar to a greyscale image and shows the brightness of the captured scene, acquired in the near infrared range.

The measured distance d of the camera to the surface of the captured objects can be calculated with the following equation:

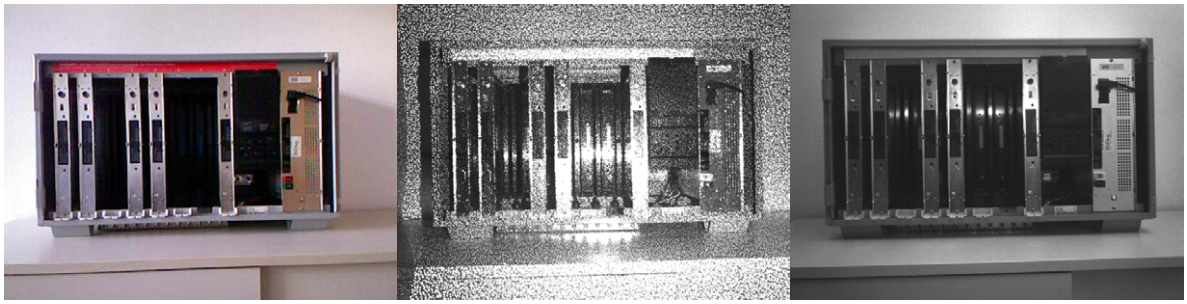
$$d(m) = \frac{c}{4\pi f} \cdot \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \tag{2.19}$$

2.2.2. Structured light depth camera

Recently, the first low-cost depth camera for the consumer mass market was developed by PrimeSense [Pri13]. This depth camera has become widely distributed among consumers as part of the "Kinect", an input device for gaming consoles. The depth sensing technology developed by PrimeSense doesn't have a custom name. Therefore, this thesis refers to it by the name of the most widely distributed physical device this depth sensing technology was built into (Kinect). However, the same depth sensing technology has been integrated in other depth imaging devices as well. For example, the Xtion depth camera is also based on this depth sensing technology. The comparative evaluation conducted by Gonzales et al. showed that other depth cameras based on the PrimeSense technology have very similar measurement properties as the Kinect [GJRVF*13].

The Kinect contains two cameras (a color camera and an infrared camera), as well as an infrared projector which projects an infrared pattern onto the scene. The baseline between the projector and the depth camera is about 75mm. The infrared pattern projected onto the captured scene is detected by the infrared camera and used to calculate the distance of the depth camera to the captured object surface. Thus, the Kinect depth camera uses a "structured light" based approach to estimate the depth. The deformation of the pattern projected onto the three dimensional objects is used to infer their three dimensional shapes [ZSMG07]. Figure 2.6a and 2.6b show the color and the infrared images of a Kinect camera. In the infrared image, the pattern projected by the infrared projector is visible. Figure 2.6c visualizes an image captured by the infrared camera while the projector was covered with opaque

material, such that no pattern is projected onto the scene. Furthermore, 2.7 shows the point pattern projected onto a planar wall. The exact working principle of the Kinect has not been published explicitly. However, PrimeSense has filed several patents closely related to the depth sensing technology used for the Kinect [ZSMG07] [SZ08] [FSA10]. This section describes the technology described by the patents. Thus, the next paragraphs are based on the assumption, that the patented technologies are used for the Kinect's depth estimation as they were described in the patents.



(a) Image of color camera (b) Image of infrared camera (c) Image of infrared camera (projector occluded)

Figure 2.6.: Kinect point pattern projected onto an object (used to calculate the depth for each pixel).

Structured light Zalevsky et al. disclosed the main depth sensing principle in a first patent [ZSMG07]. This patent describes the structured light based approach, which is used to estimate depth based on the distortion of the projected pattern. The projector of the Kinect projects a non-periodic speckle pattern onto the captured scene. Then, an infrared image of the projected pattern is acquired. The Kinect requires only a single image of the projected pattern to estimate the depth.

After the acquisition of the infrared image, the depth is estimated by an embedded control unit, which stores a reference image of the projected pattern. To calculate the depth, the control unit analyzes the captured image to determine a shift of the pattern in the image of the object relative to the reference image. A correlation based image matching algorithm is used to calculate the shift of the projected pattern. The control unit estimates a depth image from each captured infrared image. Using the embedded control unit, a new depth image is calculated 30 times per second, providing real-time depth imaging.

Shape characteristics that change with distance (astigmatic lens) A subsequent PrimeSense patent, filed by Shpunt et al. [SZ08], describes a method for depth image estimation that combines shape-based ranging with shift-based mapping. The transverse shifts of parts of the projected pattern are used to reconstruct the depth values. Just as described in the first patent [ZSMG07], these shifts are compared to the reference pattern of the projection at a known distance. The major claim of this subsequent patent is a method for mapping, which comprises the projection of a pattern of multiple spots onto an object

with the following properties: the positions of the spots in the pattern are uncorrelated, while the shapes share a common characteristic [SZ08]. Furthermore, the patent claims spot shape characteristics which change with varying projection distances. Thus, the shapes of the spots on the surface of the captured objects can be used for the depth estimation. This distance-varying shape is created by passing the projection beam through optical elements, which split up the beam into multiple spots and which create the distance-varying shape [SZ08].

An optical element for creating projected shapes that vary with the projection distances is described in a third patent [FSA10]. An astigmatic optical lens is used for this purpose. This astigmatic lens has a different focal length in the horizontal direction than in the vertical direction. Thus, it elongates the projected speckles: with such an astigmatic lens, projected circles become ellipses [Mac11]. Furthermore, the direction of elongation varies with the distance of the projection. Thus, depths can be estimated by analyzing the directions of the elongations.

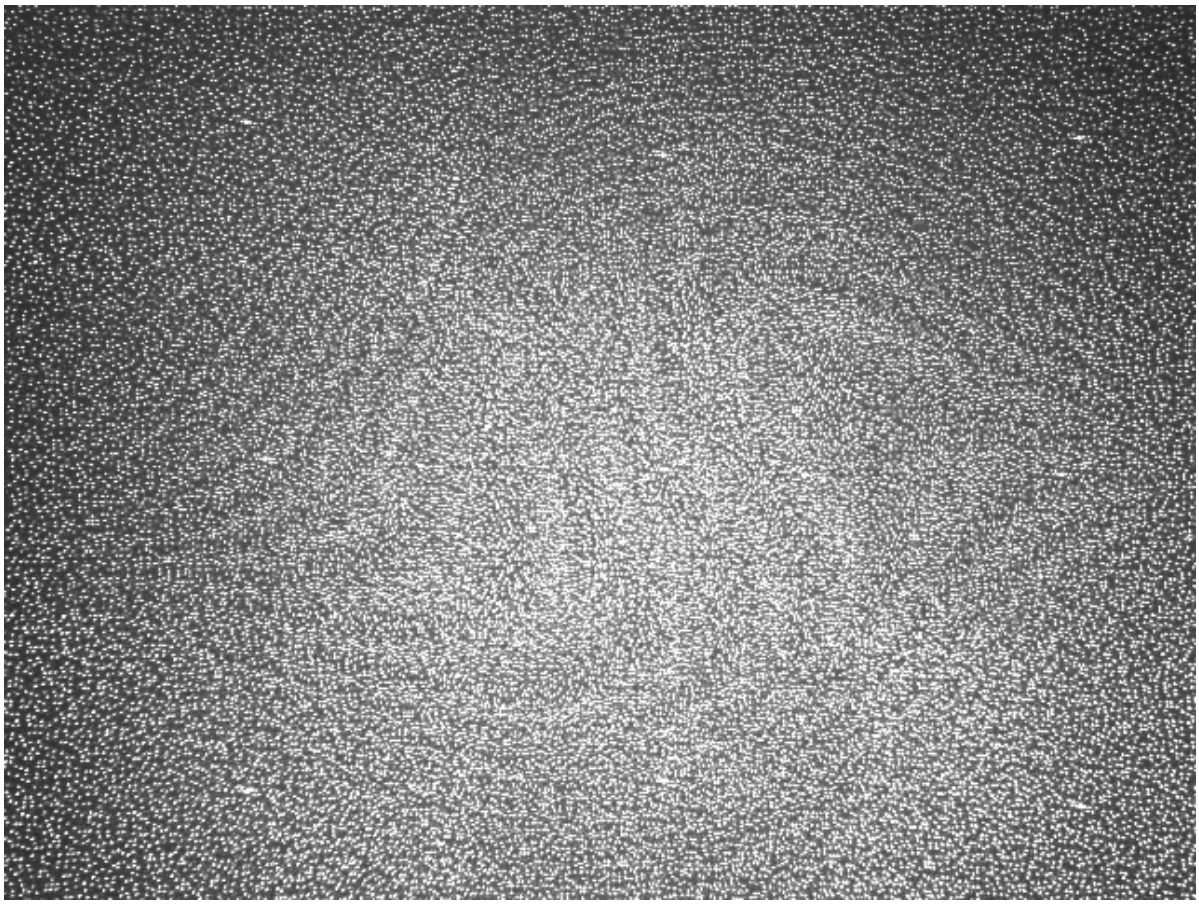


Figure 2.7.: Kinect point pattern projected onto a flat wall.

2.2.3. Measurement errors of depth cameras

The depth measurements of depth cameras are affected by random noise as well as by systematic errors. Due to random noise, the measured distances differ slightly from frame to frame, even for a static scene and a static camera position. For state-of-the art time-of-flight cameras, the random measurement noise is typically stated to be about 1% of the measuring distance [Mes13]. The random measurement noise of a Kinect is significantly smaller for close measurements and increases quadratically with the distance, up to 0.8% at the Kinect's maximal measuring distance (5 m) [KE12]. The random measurement noise of depth cameras can be modelled with a Gaussian distribution. In addition to the random measurement noise, systematic measurement errors reduce the 3D measuring accuracy. They are specific for each depth sensing technology.

Time-of-flight cameras Several studies provide error analyses of the measurement errors of time-of-flight depth cameras [FH08] [RFHJ08] [LSKK10] [FAT11] [BBGB*12]:

- **Wiggling error ("depth distortion")** Due to inaccuracies in the modulation process of the emitted light, the light emitted by time-of-flight cameras is not perfectly sinusoidal. Therefore, wiggling errors cause systematic under- and overestimates of the distance, in relation to the distance of the camera to the measured object surface.
- **Integration-time related errors** The integration time of a depth camera is the time during which the reprojected light is collected by the sensor. If the integration time is short, the amplitude of the received signal is low. In this case, only few photons get projected to each gate of the sensor, which reduces the measurement accuracy. On the other hand, if the integration time is too long, 3D measurements cannot be acquired due to oversaturation.
- **Built-in pixel-related errors** Material properties of the sensor cause distance measurement offsets for each pixel of the sensor.
- **Amplitude-related errors** If only few photons get reflected to a pixel of the sensor (i.e., if the amplitude of the amount of reflected light is low), the signal-to-noise ratio and thus the measurement accuracy decreases. Even if the integration time is appropriately set, low amplitudes cannot be avoided. In addition to the integration time, there are three main causes for low amplitudes [FAT11]. First, the illumination emitted by time-of-flight cameras is brighter near the image center than at pixels that are more distant from the image center. Therefore, the amplitude (and thus the 3D measurement accuracy) decreases with increasing distance from the image center. Second, objects in the captured scene which are further away are less brightly illuminated than objects close to the camera. Third, the amplitude decreases if the material surfaces have a low reflectivity or for large angles between the viewing direction and the surface normal.
- **Temperature-related errors** Due to the sensitivity of the depth camera's semiconductors to temperature changes, the 3D measurements acquired by time-of-flight cameras are influenced by the working temperature of the sensor. Thus, the acquired 3D measurements drift when the temperature of the camera changes.

- **Multiple light reception & multiple reflection paths** These measurement errors occur when light emitted by the time-of-flight camera, which got reflected by different parts of the captured scene, is projected onto the same pixel of the depth camera's sensor. For example, these measurement errors occur at the corner of two objects. Figure 2.8 visualizes an overestimation of the distance due to the reprojection of light from two different surfaces. Here, the light reprojected onto the sensor is not only from the direct path, but also from a multiple reflection path.

  Due to multiple light reception, the measurement accuracy of time-of-flight cameras at the edges of objects is very poor: at edges, light gets reflected both from the foreground object and from the background. Thus, the measured distance is an intermediate value between the distance to the foreground and the distance to the background object.
- **Motion blur** Motion not only blurs 2D images, but reduces the accuracy of depth measurements as well.

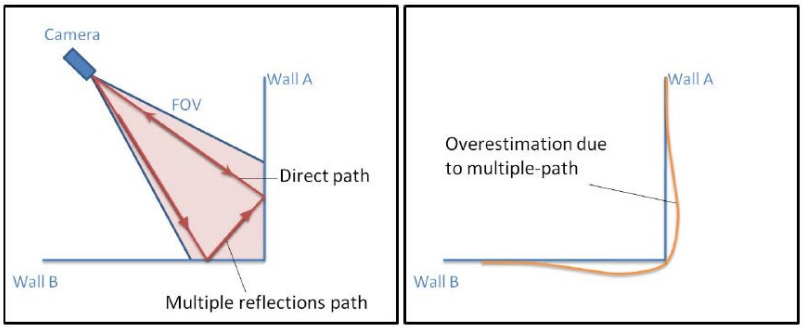


Figure 2.8.: Multiple path reflections (from SwissRanger 4000 manual [Mes13]).

Kinect Most of the measurement errors of time-of-flight cameras do not affect the structured light based approach (neither the wiggling error, integration-time related errors, amplitude-related errors nor multiple light reflections or multiple reflection paths).

However, Chow et al. [CALT12] showed that the measurement accuracy of the Kinect is subject to temperature related errors as well. Furthermore, the depth values provided by a Kinect are discretized to one of 2048 depth values (11 bit). Therefore, the depth resolution decreases quadratically with the measurement distance. It is about 2 mm at 1 m meausuring distance, 25 mm at 2.5 m distance and 70 mm at this depth camera's maximal measuring distance of 5 m [KE12].

Both time-of-flight cameras and the Kinect structured light depth camera cannot acquire 3D measurements at pixels of the captured image that are overexposed. Furthermore, with the structured light approach, 3D measurements cannot be acquired at regions in which the projected point pattern is occluded. These occlusions occur due to the different positions of the infrared projector and the infrared camera, which are separated by a spatial baseline (the black pixels in the difference visualizations of the raw Kinect depth images in Chapter 5.2 represent missing 3D measurements).

2.3. State of the art: difference detection

In 1964, Shepard was one of the first to emphasize the need for a device for automatic change detection [She64]. He pointed out that without automatic difference detection, the manual visual comparison conducted by a human risks to be slow, tiring and subject to errors. Therefore, Shepard stated the need for a device "which will automatically correlate and compare two sets of photography and indicate all the changes". Furthermore, he emphasized the need for a clear visualization of automatically detected changes: "Changes must be prominently displayed, so they will not be overlooked and no time will be wasted in finding them" [She64].

The concept of change detection is closely related to the detection of differences between two objects. The difference between both research areas is, that in contrast to difference detection between two objects, change detection refers to a single object which changes over time [Ren02]. However, often algorithmically similar methods can be applied to compare an object at several time instances and to compare two different objects. Therefore, this section describes the state of the art in image based difference detection and in change detection.

2.3.1. Difference detection with 2D images

First, the related work about difference detection between 2D images is described in this subsection. Then, approaches which use 3D input data (sometimes combined with 2D images) are described in Section 2.3.2.

Optical devices The first approaches for image based difference detection were developed for detecting differences between static 2D images. First solutions for supporting humans in difference detection tasks were proposed when computers were not yet widely-used. Thus, the first approaches for 2D difference detection were not implemented algorithmically. Instead, optical devices were used to superimpose photographic transparencies, taken on slide film [LC86]. In 1976, Ebersole described a triangular interferometer consisting of light sources, of two mirrors and of a beam splitter [EW76]. The interferometer superimposes the beams illuminating the transparencies of the two photographs. Thus, photographic transparencies can be superpositioned and subtracted in real time. Ebersole used this optical device to detect differences between photographs acquired with NASA Landsat satellites.

Similarly, Chao described an image difference detection system which also uses light passing through two transparencies [CL90]. In this setup, light passes through two coplanar image transparencies and is projected onto a common image plane such that the two image projections are coincident. The light that passes through the transparencies is polarized in transverse directions and then brought into concidence with a Wollaston prism. Due to the different directions of the polarization of the light beams, they are 180° out of phase. Thus, the light beams of parts of the images which are identical interfere and cancel out one another. Therefore, only those projected parts of the image transparencies become visible on the common projection plane which differ from each other.

Aerial photography and remote sensing One of the most important application areas of change detection is the analysis of data aquired with aerial photography and with remote sensing. Differences between captured images are analyzed in order to detect changes of urban structures, for example caused by urban expansion. There are two main challenges for detecting changes in aerial photography. First, the photographs can be taken from noncoincident positions. Second, differences caused by shadows, clouds and seasonal differences of vegetations need to be distinguished from changes of urban structures [She64] [Ald79]. While optical devices can assist in the difference detection task in general, the algorithmic analysis of digital images provides better means to differentiate between relevant and irrelevant changes, for example caused by shadows. Thus, when computers and digital images became more widely distributed, optical devices were replaced with algorithmic approaches for difference detection between 2D images [Sin89].

With images acquired from aerial photography and remote sensing, changes can either be detected between raw captured images, or between images with hand-labeled regions [RAAKR05]. Algorithms for change detection in 2D images either apply supervised approaches (using a reference set with ground truth classification data for learning) or unsupervised approaches [PNPNGLFR05]. These algorithms label all regions of the captured image according to detected changes. The labeling can either be discrete ("changed" or "unchanged"), or a probability value denoting that detected differences reflect a change.

Supervised classification approaches can were for example used to detect changes of buildings for cartographic purposes [BI12]. Algorithms commonly used for supervised classification are post classification comparison, vector machines or neural networks [DLG*12]. However, supervised classification approaches have the drawback that an existing database is required to learn classes that distinguish the requested structures (such as buildings) from other urban structures. Thus, they require prior knowledge.

In contrast to supervised approaches, unsupervised approaches do not require reference data for learning classification sets. Algorithms that can be used for unsupervised change detection in the context of aerial photography and remote sensing are: image differencing, regression, principal component analysis and independent component analysis [DLG*12].

If image regions of aerial photographies and other data acquired by remote sensing are classified as changed or as unchanged, both omission and commission errors occur. In order to reduce both kinds of errors, Du et al. proposed to fuse several difference images [DLG*12]. They used a fuzzy set theory fusion model to calculate the probability of change of each pixel by analyzing the uncertainty of each input image. Such a data fusion approach combining data from several images can reduce the inaccuracies that would occur if only a single input image would be used for the change detection.

Construction planning In the context of construction planning, several approaches have been proposed which assist humans with the manual detection of differences [GSB*07] [GBSN09] [SS08] [FG11]. These approaches do not calculate the differences algorithmically. Instead, in order to ease the manual difference detection task, 2D images of the real scene are visually superimposed with CAD

planning data. It is then up to the user to detect differences between the 2D images and the superimposed visualization of the CAD planning data.

The main challenge of these approaches is a correct alignment of the CAD planning data with the captured 2D images. An appropriate alignment is required to visualize the CAD data with the correct size, position and orientation. This task is solved with augmented reality techniques, which use camera tracking algorithms to estimate the position and orientation of the camera in relation to the scene. Given the position and orientation of the camera, the 2D camera image can be superpositioned with the CAD data such that the CAD data is correctly aligned with the 2D image.

Georgel et al. proposed an augmented reality solution for difference detection in the context of construction planning [GSB*07] [GSN09] [GBSN09] [FG11]. They use the term "discrepancy check" for the detection of differences between the planned 3D model and the built object. The discrepancy check is used to validate the 3D model. The system proposed by Georgel et al. allows engineers to superimpose still 2D images of a plant with the CAD model developed during the planning phase. The proposed system uses an offline approach: the differences are not detected directly on site. Instead, the user first takes several photos of the relevant object on site, before returning to the office. There, the recorded photographs are uploaded to a database and the 3D model is aligned with the recorded photographs. In the approach proposed by Georgel et al., rectangular structures installed on the walls of industrial buildings are used for the alignment. To align the 3D model with the photographs, the user matches segmentations of the anchor plates from the recorded photographs with corresponding anchor plates in the CAD model.

Schoenfelder and Schmalstieg proposed a system for documenting changes between the planning documentation of an industrial building and the as-built status of the building [SS08]. Similar to the system proposed by Georgel et al., this system superimposes CAD planning data with images of the building. However, in contrast to the system proposed by Georgel et al., this system allows to detect differences directly on site. They use a wheel-mounted mobile AR device with a touch screen and a camera, which can be moved along the floor of the factory. The camera is rigidly coupled with the mobile device, so its height above the floor remains constant. The pose of the camera on the mobile device is tracked with outside-in optical camera tracking. Therefore, an external four-camera setup is installed around the wheel-mounted mobile AR device. This external multi-camera setup estimates the pose of the mobile device by calculating the positions of spherical markers attached to the mobile device. In order to align the CAD data with the real factory, 3D points are selected on the CAD model. Then, the 3D positions of these 3D points in the real factory are measured with geodetic surveying and markers are attached to these 3D positions in the real factory. By detecting these attached markers in the camera images of the external multi-camera tracking system, the coordinate system of the CAD data is aligned with the real factory.

Just as the approach proposed by Georgel et al. [GSB*07] [GBSN09], the system described by Schoenfelder and Schmalstieg [SS08] provides augmentations of the CAD data with images of the real object. However, the surface of the real object is not measured in order to algorithmically compare the surface of the real object with the surface of the CAD data. Instead, it is up to the user to detect differences by manually comparing the visualization of the CAD data with the images of the real object.

Video encoding In the context of video encoding, change detection has been applied to improve the data encoding. Lee et al. proposed a method to automatically adjust the data encoding rate of a video [LSP*09]. Therefore, they divide the current frame into several regions and calculate a dissimilarity metric for each of these regions. If the dissimilarity of several regions is above a threshold, this is interpreted as a scene change and the encoding rate is adjusted accordingly. Similarly, as part of the video encoding process, Jiossy et al. detect scene changes prior to motion estimation and to intraframe prediction, in order to reduce motion prediction overhead and to support the structuring of groups of pictures [JR11]. Changes are detected by analyzing image histograms of subsequent frames. Jin et al. use difference detection to reduce the computational complexity of H.264 video encoding [JG11]. To detect differences, they analyze the color and moving correlation features of the video. Then, the occurence of changes is taken into account for distributing the video data to different modules of the video encoder.

The different approaches trackling change or difference detection for video encoding have in common, that they divide the frames of the videos in separate regions and that they calculate for each region, whether this region has changed since previous frames [LSP*09] [JR11] [JG11]. Thus, for video encoding, the question is whether anything has changed or not. In the context of video encoding, this is equal to the question whether the average color change in a certain region is larger than a threshold.

A more complex approach for detecting differences in videos was proposed by Pickup [PZ09]. In this work, an approach for detecting visual continuity errors in movies is described. Visual continuity errors are parts of the scene's background, such as the position of an object, that might vary unintentionally in several recorded takes of the same scene. To detect continuity errors, first a point-to-point registration between image pairs of a scene is calculated. The registration is estimated with a planar projective transformation. This approach is based on the assumption that the distance to the background objects of the observed scene is far enough to be approximated by the distance to a common plane. Using the estimated registration, a discrepancy score is calculated for each pixel by comparing the local neighbourhood of each pixel in a frame with the local neighbourhood of the corresponding pixel in the other frames. Finally, detected differences in the background are distinguished from changes caused by movements of the actors. Therefore, the actors are detected with a human upper-body detector.

2.3.2. Difference detection with 3D input data

Approaches for difference detection with 3D input data have been proposed in the context of urban change detection, version control systems, CAD model comparison, 3D reconstruction and for the comparison of as-built with as-planned data.

Aerial laser scanning and remote sensing As an extension of two dimensional aerial photography, remote sensing technologies provide 3D measurements as well. Such 3D data can either be acquired with airborne laser scanners or by calculating depth from stereoscopic aerial images. Just as for the 2D image based change detection approaches described in Section 2.3.1, 3D urban change detection is solved by object detection and by object classification. For the object detection and object classification,

similar algorithms are applied as for 2D image based urban change detection. For example, Xiao et al. detected trees in 3D datasets acquired in different years, in order to estimate the growth of the trees [XXOEV12]. Malpica et al. and Stal et al. proposed algorithms to detect buildings in 3D data, in order to estimate changes in urban structures [MAP*13] [STDM*13].

Comparing two virtual 3D models with version control systems Difference detection between two 3D models is investigated in the context of version control systems. In contrast to the approach described in this work, the geometric shape of a 3D model is not compared to the shape of a real object, but to another virtual 3D model. Similar to version control systems for text documents (such as subversion or GIT), specific version control systems for 3D models have been proposed [ASW09]. When an updated version of a 3D model is committed, these systems try to detect whether the submitted changes conflict with the most recently submitted version of the 3D model. Therefore, 3D model version control systems either compare textual lists of changes ("change-based comparison") or the state of two 3D models and their common ancestor ("state-based comparison"). State-based comparisons either rely on universally unique identifiers (UUIDs) or use structural analysis to detect changed parts of the 3D models. This task is eased by specific properties of difference detection by version control systems: first, the compared 3D models have a common ancestor. Thus, it is not necessary to find out which part of the first model corresponds to which part of the second model. Furthermore, version control systems usually just provide a binary decision whether a part of the 3D model was changed or not. For example, Dobos and Steed recently developed a tool for differencing and merging 3D models [DS12]. If a single vertex was deleted or repositioned, the entire node which contains this vertex is considered different.

3D CAD model comparison and 3D shape retrieval Elaborated methods for comparing the actual shape of two 3D models were developed in the context of 3D CAD model comparison [BCRM12] [BCRM13] and 3D shape retrieval [YHY07] [TV08]. 3D CAD model comparison, 3D shape retrieval and the problem investigated in this thesis (difference detection between a real object and a 3D model of this object) have in common that the compared shapes need to be geometrically aligned for detecting 3D differences. Geometric alignment is the task to find a transformation such that the compared objects have the same position and orientation in 3D space (see Section 2.1.4). Briere et al. pointed out that for 3D CAD model comparison, the alignment only needs to be carried out once before comparing the two CAD models [BCRM12]. In contrast, shape based retrieval is a 1-to-N problem: To retrieve a 3D shape from a database, many 3D models in the database need to be searched to find the 3D model which best matches the provided shape. Therefore, due to the required computation time, precise but computationally intensive alignment approaches are less suited for 3D shape retrieval than for 3D CAD model comparison.

A common approach for geometric alignment in the context of 3D shape retrieval is to normalize the position and orientation of the 3D models. Therefore, first the centroids (centers of mass) of the 3D models are translated to the same position. Then, the orientations of the 3D models are normalized with principal component analysis. To normalize the orientations, the principal axes of a 3D model are

aligned to the axes of a canonical coordinate system by calculating the eigenvectors and the resulting diagonal matrix of the eigenvalues.

For real-time 3D difference detection, only one 3D model needs to be compared with the 3D measurements on the surface of the real object. However, the point of view of the 3D difference detection can change every frame, up to 30 times per second. To comply with the real-time requirement, the alignment of the captured 3D measurements on the surface of the real object and the virtual 3D model needs to be estimated very efficiently. The principal component based registration of two virtual 3D models is not applicable for registering a 3D model with a captured depth image of a real object: A depth image captures only a partial view of a real object, so the position and orientation of the real object can not be normalized with principal component analysis.

Robotics (3D change detection) In robotics, a 3D model of the environment is commonly reconstructed for navigational tasks, i.e. in order to estimate the position and orientation of the robot in a previously unknown environment. Changes in the environment that occur after the 3D reconstruction can either be handled implicitly (by continuously updating the 3D reconstruction, without explicitly calculating the observed changes), or explicitly [AML07] [NDB*10] [VDC12].

Nunez et al. studied the problem of novelty detection, which is closely related to change detection [NDR*09] [NDB*10]. They define novelties as perceptions which were not experienced before. A laser installed on a mobile robot is used to acquire 3D point clouds of the environment. To detect novelties, a robots' current 3D measurements are compared with a 3D point cloud previously acquired by the robot. Nunez et al. detect clusters of previously unobserved 3D points with probabilistic Mixtures of Gaussian functions. In order to represent the detected novelties, the detected clusters are then matched to one of three predefined primitive shapes (spheres, cylinders and planes). Similarly as Nunez et al., Vieira et al. also cluster 3D points in order to detect changes in 3D point clouds acquired by a mobile robot [VDC12]. However, instead of Mixtures of Gaussian functions, they use implicit volumes in order to cluster the 3D point clouds. First, both the reference point cloud and the changed point cloud are converted to implicit volumes. Then, Vieira et al. apply boolean operations to the 3D data to detect changes. This approach has the advantage that the shapes of the detected objects are not limited to specific primitives.

The approaches proposed by Nunez and Vieira do not explicitly address the problem of accurate registration of the 3D point clouds. Both 3D points are coarsely aligned by the estimation of the position and orientation of the robot. However, inaccuracies in the localization of the robot limit the overall accuracy. The approaches proposed by Nunez and Vieira can be used to detect large new objects, such as a human wo entered a previously empty corridor, or a large box placed on the floor. Both approaches are too slow for update rates of several frames per second, as they require several seconds up to several minutes for the calculation of changes.

Andreasson et al. proposed an approach for robotic difference detection which does not rely on the assumption that the robot maintains a consistent coordinate system between several 3D scans of the environment [AML07]. They use a robot which acquires both 2D color images and 3D laser scan data.

In order to align new 2D and 3D data with previously collected data, they first detect SIFT features in the 2D color image. These visual features are used to establish correspondences between 3D points within the entire 3D point cloud. First, they detect the scan in the reference model that is most similar to the current scan. Then, they use the 3D points which correspond to the 2D positions of the SIFT features to align the current scan with the reference model. The reference model is represented with a 3D normal distribution transform. Therefore, the 3D space of the reference 3D measurements is divided into cells. Then, for each cell, the mean value and the covariance is calculated from all 3D points within the cell. The probability that a measured 3D point is different from the reference data can then be computed with the mean value and the covariance of the cell that contains the measured 3D point. The system described by Andreasson et al. uses a pan-tilt unit mounted on the robot which is sweeped to acquire laser scan data and color images with different orientations. Thus, the 2D and 3D data is not acquired at interactive frame rates. Thus, for this system, there is no real-time constraint for calculating the differences.

The different approaches for robotic change detection have in common that they detect changes between current 3D measurements and previously acquired 3D measurements. Thus, these approaches are well suited for self-reconstructed 3D models. However, they do not handle the problem how to detect differences between 3D measurements and arbitrary 3D models. While 3D models can be represented by arbitrary data formats, self-reconstructed 3D models can be adapted to the task at hand. Thus, it is easier to detect differences between 3D measurements and self-reconstructed 3D models than between 3D measurements and arbitrary 3D models.

Difference detection at a single 3D point Webel et al. proposed a system for augmented reality based discrepancy check with which the 3D positions of single points in the 3D model and the real scene can be compared [WBSW07]. This approach uses a stereo camera system and a hand-held laser pointer. The laser pointer is used to depict a point on the surface of the real scene. The 3D coordinate of the point is reconstructed by triangulation with a stereo camera system. While this approach allows the comparison of single 3D points, it is not suited for dense 3D difference detection.

Comparing the as-planned data with the as-built status of buildings In order to detect differences between CAD data and 3D measurements, Bosche et al. transformed the 3D data of a CAD model and 3D data measured with a time-of-flight depth camera into a common voxel occupancy grid [BTHC06]. As they used a static camera position, they did not need to solve the registration problem for moving camera positions.

Several approaches for 3D difference detection have been proposed for the comparison of the as-planned status and the as-built status of buildings which use stationary laser scanners. The setup of such a laser scanner typically takes several minutes. Furthermore, using a laser scanner which sequentially scans the environment, a single 3D scan of the environment takes about one to three minutes [ABG*06] [TAH09]. The position of such laser scanners can not be changed during the data

acquisition. However, laser scanners provide very precise 3D measurements, with mm-level accuracy at a measurement distance of several meters.

Tang et al. proposed an approach for detecting surface flatness defects in laser scan data [TAH09]. In their work, the detection of flatness defects is eased by the fact that they exclusively inspect a planar surface. Thus, 3D differences can be detected by fitting the acquired 3D point cloud to a plane. Then, 3D measurements which are not close to the calculated plane are potential candidates for flatness defects.

Detecting differences between CAD data and 3D laser scan data for objects with non-primitive shapes (i.e. for objects that are more complex than a plane) requires a registration between the CAD model and the 3D measurements. Bosche addressed the registration problem in the context of object recognition for the comparison of as-planned with as-built data [Bos08] [Bos10]. He used manually specified 3D correspondences between a laser scan of a construction site and a 3D model of the construction site to transform both data sets into a common coordinate system. This manual alignment had to be repeated whenever the 3D laser scanner was moved to another position. As the manual selection of 3D correspondences is an offline step (and for the task of discrepancy check, 3D-3D correspondences can not easily be extracted automatically as the 3D model might differ from the real scene), this approach is not feasible for difference detection with an arbitrary moving camera.

Akinci et al. describe three possible registration methods that could be used instead of a manual registration [ABG*06]. First, the 3D laser scanner could be tracked with outside-in pose estimation. Therefore, an external tracking system could be used to estimate the position and orientation of the laser scanner. Second, markers attached to known locations of the real scene could be detected by the scanner. Third, the pose of the sensor could be calculated with vision based approaches for the registration of several 3D scans. However, Akinci et al. also use manual registration, both for the pose estimation of the laser scanner and for the registration of the as-planned data with the captured as-built data [ABG*06]. Similarly, Anil et al. detect differences between laser scan data and a 3D building information model by measuring the 3D position of the laser scan data before a scan is performed [ATAH13]. Tang et al. proposed another solution for documenting the as-is state of buildings with laser scan data [TAAH11]. However, they do not address the registration problem. Instead, they impose the requirement that all 3D measurements must be in the same coordinate system as the 3D model. They state that is requirement is fulfilled in their project because the 3D point clouds acquired with the laser scanner are first transformed to a geographic coordinate system. Then, the 3D models are created by engineers within this coordinate system.

2.4. Conclusion

This chapter provided the background for 3D difference detection. First, computer vision concepts and notations were introduced in Section 2.1. This section introduced the perspective camera model, transformations between different coordinate systems, depth images and 3D point clouds, as well as the concepts of registration and alignment.

Then, approaches and devices for capturing 3D measurements were described in Section 2.2 and in Section 2.2. This description focused on depth cameras, which acquire dense depth images (respectively dense 3D point clouds) in real time. Therefore, the 3D data acquisition principles of the two main kinds of real-time 3D depth cameras were described: time-of-flight depth imaging and structured light based depth imaging.

Finally, this chapter provided an overview of the state-of-the-art in difference detection. While real-time, depth image based 3D difference detection for arbitrary 3D models and moving camera positions has not been described previously, difference detection has been studied in several related research areas. Therefore, in Section 2.3, solutions for detecting differences in 2D images were described as well as approaches that detect differences with 3D input data. The remainder of this chapter summarizes the applicability of the described state-of-the art approaches for real-time 3D difference detection with an arbitrary moving depth camera.

Applicability of previous difference detection approaches for real-time 3D difference detection The previously proposed approaches for difference detection can not be used for real-time 3D difference detection with an arbitrary moving camera. None of the previously described approaches fulfills all required criterions for real-time 3D difference detection with an arbitrary moving camera. The previously described approaches are not applicable for this task due to the following limitations:

- The solution proposed by Webel et al. can only be used to calculate 3D differences for single 3D points [WBSW07].
- Other previous solutions for difference detection do not calculate 3D differences at all [GSB*07] [SS08] [GBSN09] [FG11]. Instead, a 2D image of the real object is visually augmented with the 3D model. Then, it is up to the user to manually detect differences by visually comparing the 3D model with the 2D image.
- The approach described by Tang et al. is restricted to the detection of differences at a single planar surface [TAH09].
- Most previously described approaches use a static camera position, respectively static laser scan positions [BTHC06] [ABG*06] [GSB*07] [Bos08] [GBSN09] [TAH09] [Bos10] [FG11]. A manual alignment step is required every time the scan position is changed. Such a manual alignment requires the selection of corresponding points on the captured 3D data and on the 3D model. Thus, these approaches are not suited for 3D difference detection for arbitrary moving camera positions. The approach described by Tang [TAAH11] furthermore imposes the requirement that the 3D model must already be aligned with the laser scan data.
- Most previously described solutions are not real-time capable [ABG*06] [AML07] [GSB*07] [Bos08] [GBSN09] [TAH09] [NDB*10] [Bos10] [TAAH11] [FG11] [VDC12].

Table 2.1 summarizes the state of the art approaches for difference detection. None of the previously proposed approaches fulfills all requirements for real-time 3D difference detection from arbitrary camera positions.

Approach	Acquisition of dense 3D measurements	Calculation of 3D differences	Scan position can be changed without re-calibration	Real-time & on site	Remarks
[GSB*07] [GBSN09] [FG11]	no	no	no	no	Visual augmentation of 2D images with the 3D model. No explicit difference calculation or difference visualization.
[SS08]	no	no	✓	✓	
[WBSW07]	no	At sparse 3D points	✓	✓	Sparse 3D difference detection at single 3D points, no dense 3D difference detection.
[TAH09]	✓	✓	no	no	Specialized approach for a planar surface, not applicable for arbitrary 3D shapes.
[ABG*06]	✓	Unclear	no	no	3D measurements acquired with stationary laser scanners. A manual re-calibration is required whenever the scan position is changed.
[Bos08] [Bos10]	✓	✓	no	no	
[TAAH11] [ATAH13]	✓	✓	no	no	
[AML07]	✓	✓	✓	no	Detection of changes between current 3D measurements and previously acquired 3D measurements. These approaches target self-reconstructed 3D models, not arbitrary 3D models.
[NDB*10]	✓	✓	✓	no	
[VDC12]	✓	✓	✓	no	

Table 2.1.: Summary of related approaches for difference detection. None of the previously proposed approaches fulfills all requirements for real-time 3D difference detection from arbitrary, moving camera positions.

3. Depth image based 3D difference detection

This chapter describes the concept for real-time 3D difference detection with a depth camera. As pointed out in Section 2.1.3 and in Section 2.2, depth cameras acquire depth images (respectively dense 3D point clouds) in real time. However, state of the art approaches for difference detection with laser scanners, 2D cameras or robots cannot be used for dense real-time 3D difference detection with a moving depth camera (see Sections 2.3 and 2.4). They either assume a static scan position, are not real-time capable or are restricted to the detection of differences between current and previous 3D measurements (as opposed to the detection of differences between 3D measurements and an arbitrary 3D model). Thus, there is a need for an approach for real-time 3D difference detection from arbitrary camera positions. Therefore, this chapter addresses the research question Q1:

Q1 ***How can 3D differences be detected in real time and from arbitrary viewpoints using a single depth camera?***

As pointed out in Section 1.3, this question can be divided into two subquestions:

Q1.1 How can the 3D measurements of a depth camera be mapped onto an arbitrary 3D model in real time?

Q1.2 Given a mapping of 3D measurements onto a 3D model, how can 3D differences be detected in real time for a moving depth camera?

The concepts described in this chapter were published in the publications [KWSF10] [FKOJ11] [KK12] [Kah13] and [KBKF13]. The question Q1.1 is addressed in the Section 3.1, by proposing a general concept for real-time 3D difference detection which builds on the fusion of computer vision and computer graphics.

Section 3.2 addresses both Q1.1 and Q1.2 by introducing the main algorithmic components for real-time 3D difference detection. Furthermore, Section 3.3 provides an answer to question Q1.2 by describing several concrete instantiations of the general 3D difference detection concept. As an amendment to the application scenarios described in the first Chapter, Section 3.4 sketches how real-time 3D difference detection can support 3D modeling and augmented reality applications.

Section 3.5 summarizes the key aspects of the proposed concept and lists the sources of inaccuracies that arise within the 3D difference detection process. Approaches for reducing these sources of inaccuracies will be proposed and quantitatively evaluated in the following chapters.

3.1. Concept

In order to address the question Q1.1, this section introduces an approach for mapping the 3D measurements of a depth camera onto an arbitrary 3D model in real time. First, the main challenges for mapping 3D measurements acquired by a depth camera onto an arbitrary 3D model in real time are described. Then, the concept of the proposed approach is introduced, which builds on the fusion of computer vision and computer graphics.

Main challenges The real-time comparison of 3D measurements acquired with a moving depth camera with an arbitrary 3D model is subject to two major challenges: first, the position of a hand-held depth camera is arbitrary and changes every captured frame. Thus, the coordinate systems of the depth camera and the 3D model need to be aligned anew for every captured frame. Second, given an alignment of 3D measurements with the coordinate system of a 3D model (provided by the pose of the depth camera relative to the 3D model), the comparison of the measured 3D points with the 3D model still requires the calculation of a 3D-3D correspondence between each 3D measurement and the 3D model.

A possible approach to obtain 3D-3D correspondences between a 3D measurement and the 3D model would be to calculate the closest 3D point on the 3D model for each 3D measurement. If the 3D model is represented by a triangle mesh, the closest 3D point can be found by calculating the distance between the 3D measurement and each triangle of the 3D model [Jon95]. Even though such approaches can be sped up with bounding volume hierarchies, they tend to be too slow for complex 3D models that contain a large number of triangles. Furthermore, with such an approach, the distance calculation depends on the mesh representation (the algorithm would need to be adapted for meshes which are not represented by triangles) and it would be necessary to parse the internal structure of the 3D model, in order to obtain all triangles. Instead, this thesis proposes an approach that is real time capable and that does not need to analyze the internal structure of the 3D model in order to estimate correspondences between the 3D measurements and the surface of the 3D model.

Proposed solution: fusion of computer vision and computer graphics On the one hand, computer vision algorithms provide means to align the 3D measurements captured with a depth camera with the coordinate system of the 3D model in real time. On the other hand, given an alignment of 3D measurements with the coordinate system of a 3D model, the comparison of the measured 3D points with the 3D model still requires the calculation of a 3D-3D correspondence between each 3D measurement and its corresponding 3D point on the surface of the 3D model. This thesis proposes to use an analysis-by-synthesis computer graphics approach to obtain such 3D-3D correspondences in real time.

Computer vision: alignment of 3D measurements with the coordinate system of the 3D model In order to align the 3D measurements captured by a depth camera with the coordinate system of a 3D model, the depth camera is rigidly coupled with a tracking device (such as another camera or a coordinate measuring machine). In a concrete instantiation, the tracking device might also be the depth

camera itself. During the 3D difference detection, computer vision based pose estimation or other pose estimation methods that are specific to the tracking device are used to calculate the pose (R,t) of the tracking device in the fixed tracking coordinate system.

The alignment of the depth cameras' 3D measurements with the coordinate system of the 3D model is split up into two separate steps. In a first offline step, which is described in Section 3.2.1, the relative transformation between the tracking device and the depth camera is calculated. Furthermore, the coordinate system of the 3D model is aligned with the tracking coordinate system.

The second step is applied in real time, during the 3D difference detection process. In this step, the pose of the depth camera is calculated from the pose of the tracking device (see Section 3.2.2). Due to the separation of the alignment into these two separate phases (offline preparation and real-time pose estimation during runtime), the pose of the depth camera relative to the 3D model can be calculated in real time.

Computer graphics: 3D-3D correspondences between 3D measurements and the 3D model In order to obtain a 3D mapping of the captured depth measurements to corresponding 3D points on the surface of the 3D model, we propose to use an analysis-by-synthesis computer vision approach which will be described in more detail in Section 3.2.3. Using the intrinsic camera parameters of the depth camera, the 3D model is rendered from the point of view of the depth camera (which were estimated with the computer vision algorithms). Then, the depth buffer is acquired from the graphics card and compared to the depth image acquired by the depth camera. As depth cameras do not acquire unordered 3D point clouds but depth images (in which each 3D measurement was acquired at a specific pixel), the captured depth measurements can efficiently be compared with the synthesized depth image with a pixel-by-pixel comparison.

Advantages of the fusion of computer vision and computer graphics for real-time 3D mapping The proposed approach can be used to compare a depth image with a 3D model in real time. Furthermore, it has the following main advantages:

- The proposed approach benefits from the great speed of the graphics card, which can handle very large 3D models in a very short time.
- Thus, the proposed approach is very fast. For example, all differences between a depth image that contains $640 \cdot 480 = 307.200$ depth measurements and a 3D model with 2.5 million triangles can be calculated and visualized in less than 15 milliseconds.
- The internal structure of the 3D model does not need to be parsed.
- This approach is applicable for any 3D model which can be rendered, irrespective of the internal representation of the 3D data.

In the next section, the main algorithmic components for 3D difference detection are described. These algorithmic components integrate the proposed 3D mapping approach into a concept for real-time 3D difference detection with a moving depth camera.

Offline preparation	**At runtime**
• Camera calibration Intrinsic calibration Depth calibration • Calculate relative transformation between tracking device and depth camera • Alignment of 3D model with tracking coordinate system	• Depth image adjustment Image undistortion Filter / adjust 3D measurements • Depth camera pose estimation • Create synthesized depth image with analysis-by-synthesis • 3D reconstruction • Pixelwise difference calculation and visualization

Table 3.1.: Main algorithmic steps of 3D difference detection (preparation and runtime).

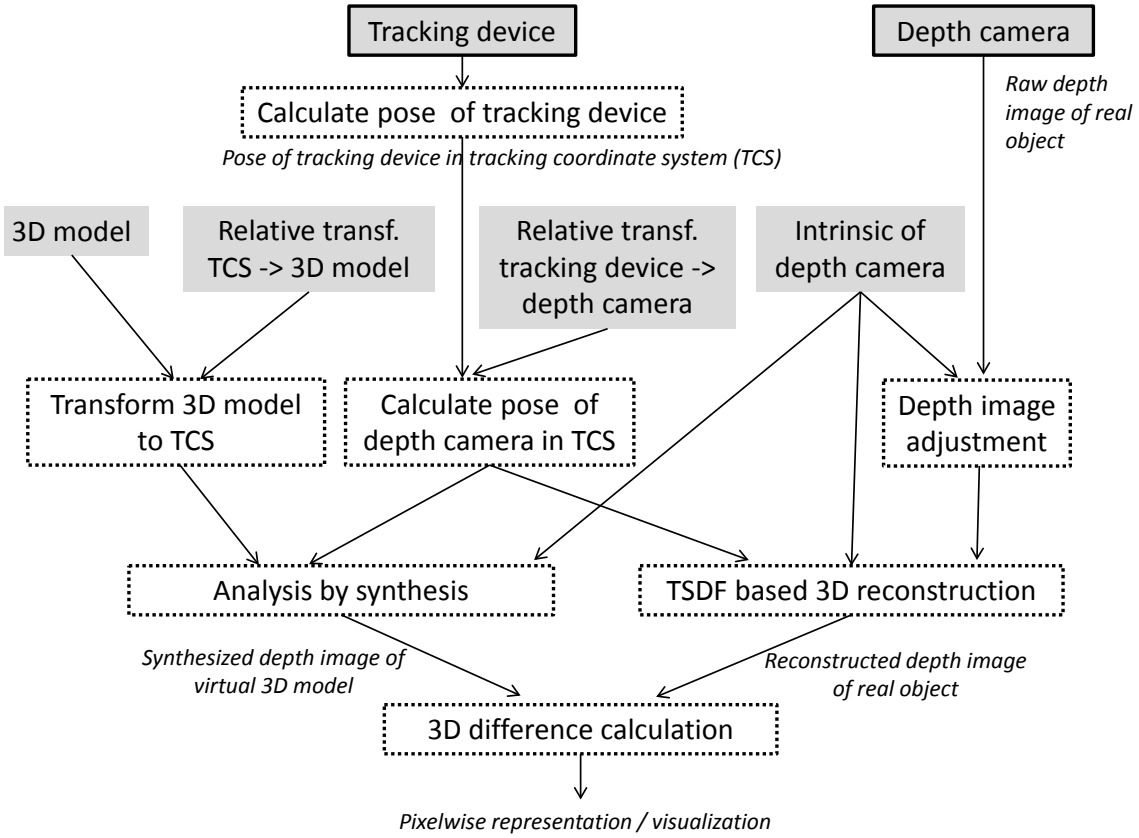


Figure 3.1.: Main algorithmic steps and data flow of 3D difference detection.

3.2. Main algorithmic components

This section describes the general algorithmic steps of the 3D difference detection. While Table 3.1 lists the main algorithmic steps, Figure 3.1 visualizes the data flow of the real-time 3D difference detection.

3.2.1. Offline preparation

In an offline preparation, the intrinsic parameters of the cameras are estimated as well as the relative transformation between a tracking device and the depth camera. Furthermore, the relative transformation between the 3D model and the tracking coordinate system is calculated, in order to align the 3D model with the tracking coordinate system.

Camera calibration The intrinsic camera parameters of the perspective camera model introduced in Section 2.1.1 are estimated in an offline calibration procedure [Cal11]. The intrinsic parameters of depth cameras (focal length, principal point, skew and tangential and radial distortion parameters) are estimated with the same algorithms as the intrinsic parameters of 2D cameras. If the depth camera is combined with a color camera, the intrinsic parameters of the color camera are calculated as well. In order to estimate the intrinsic paramters, an image sequence of a known calibration pattern, such as a checkerboard pattern with known size, is captured. In a next step, the calibration patterns' corner points are detected in the 2D image and the 3D points are projected onto the 2D image. Then, the intrinsic parameters are estimated by optimizing the intrinsic parameters such that the reprojection error between the projected 3D points and the detected 2D corner points is minimized.

In addition to the intrinsic calibration, 3D calibration algorithms have been proposed which target the reduction of systematic measurement errors of depth cameras. These approaches capture depth measurements in an offline calibration procedure and compare the captured measurements with ground truth distances, in order to derive functions that can adjust the captured depth measurements at runtime. 3D calibration algorithms for distances measured by depth cameras will be described in Section 5.1.1.

Relative transformation between the tracking device and the depth camera In order to align the depth measurements of the real object with the coordinate system of the virtual 3D model, the position and orientation of the depth camera needs to be estimated during the 3D difference detection. The device which is used to estimate the pose of the depth camera is refered to as "tracking device". The tracking device can either be the depth camera itself, or an additional device, such as another camera or a coordinate measuring machine.

If an additional device is used for the pose estimation of the depth camera, the relative transformation $(\Delta R, \Delta t)$ between the tracking device and the depth camera needs to be estimated in an offline step. The calculation of the relative transformation between two rigidly coupled cameras is called stereo calibration. If the tracking device is another camera, the calculation of the relative transformation between the depth camera and the other camera can be performed with variants of stereo calibration

algorithms [SBK08]. On the other hand, if the tracking device is a robot or a coordinate measuring machine, the relative transformation between the tracking device and the depth camera is called hand-eye transformation [TL88] [SH06]. Approaches for estimating the hand-eye transformation between a coordinate measuring machine and a depth camera will be detailled in Section 4.2.

Tracking coordinate system The term "tracking coordinate system" refers to the coordinate system in which the position and orientation of the depth camera is estimated. In this thesis, the world coordinate system is defined such that it is equal to the tracking coordinate system. Thus, both terms are used interchangeably. While "tracking coordinate system" is more specific, "world coordinate system" is the more general term.

Relative transformation between the 3D model and the tracking coordinate system In order to align the 3D model with the tracking coordinate system, the relative transformation $(\Delta R_M, \Delta t_M)$ between the tracking coordinate system and the coordinate system of the 3D model is estimated as part of the offline preparation. Then, during runtime, the 3D model can be brought into coincidence with the tracking coordinate system by attaching the 3D model to a $(\Delta R_M, \Delta t_M)$ transform node of the scene graph or the rendering system.

The relative transformation between both coordinate systems is calculated from a set of 3D-3D correspondences. Each such 3D-3D correspondence maps a 3D point from the tracking coordinate system to its 3D position in the model coordinate system. From such a set of 3D-3D correspondences, $(\Delta R_M, \Delta t_M)$ can be calculated with singular value decomposition [Ume91].

If the tracking device is a robot arm or a coordinate measuring machine, the 3D points from the tracking coordinate system can be acquired with contact based 3D measurements (see Section 4.2). On the other hand, if a contactless tracking device is used (such as a 2D camera rigidly coupled with the depth camera), the positions of the 3D points in the tracking coordinate system can be reconstructed by triangulation and an optional bundle adjustment refinement step [WWK11].

Given a set of 3D points in the tracking coordinate system, their 3D counterparts on the surface of the 3D model can either be detected automatically, by a manual selection or by a combination of both. An automatic detection can be conducted by calculating a transformation which minimizes the distances between the 3D points from the tracking coordinate system and the surface of the 3D model (see Section 4.2.2). Such a transformation can be calculated with the Iterative Closest Points algorithm [BM92] [RL01] [CSK05]. However, such an automatic alignment risks to align 3D points from the tracking coordinate system with corresponding 3D points on the surface of the model at parts of the 3D model which differ from the real object. This would cause improper or inaccurate alignments.

Thus, for the task of 3D difference detection, a semi-automatic, supervised approach can avoid this drawback of the unsupervised automatic approach. In such a semi-automatic approach, the user can make sure that 3D-3D correspondences are created on parts of the 3D model and the real object which do not differ significantly. Then, these correspondences can be refined by an automatic alignment step which minimizes the distances with the Iterative Closest Points algorithm.

3.2.2. Pose estimation of the tracking device and the depth camera

The pose of the depth camera can either be estimated based on the data captured by the depth camera itself, or by an additional tracking device rigidly coupled with the depth camera. If an additional device is used for the pose estimation of the depth camera, the tracking device first estimates its own pose (R_1, t_1) relative to the tracking coordinate system. Then, with Equation 2.9, the pose (R_2, t_2) of the depth camera can be calculated from the pose (R_1, t_1) of the tracking device and the relative transformation $(\Delta R, \Delta t)$ between the tracking device and the depth camera. Here, $(\Delta R, \Delta t)$ is the relative transformation between the tracking device and the depth camera, which was calculated in the offline calibration step.

In contrast, if the depth camera is used as tracking device, the pose (R_2, t_2) of the depth camera relative to the tracking coordinate system is estimated directly.

3.2.3. Analysis-by-synthesis 3D mapping algorithm

This algorithm maps each depth measurement captured by the depth camera to the corresponding 3D point on the surface of the 3D model. Based on the alignment of the 3D model coordinate system with the tracking coordinate system and the known pose of the depth camera relative to the tracking coordinate system, a mapping of each depth measurement to the 3D model is acquired with an analysis-by-synthesis algorithm. This approach is based on the property that the depth camera does not acquire unordered 3D measurements, but structured depth images. Furthermore, it exploits the efficiency of the graphics card to process even very complex 3D models in a very short time (a few milliseconds). Thus, with this analysis-by-synthesis approach, the depth measurements can be mapped to the 3D model in real time.

First, the 3D model is rendered from the current camera pose with the intrinsic and extrinsic parameters of the depth camera. The extrinsic camera parameters compose the modelview matrix of the rendering pipeline. In contrast, the projection matrix is defined by the intrinsic parameters of the depth camera (focal length and principal point).

In order to get a simulated depth image, the depth buffer of the graphics card and the rendering pipeline is enabled before the 3D model is rendered. The 3D model is transformed to the tracking coordinate system by attaching the 3D model to a root node which models the relative transformation between these two coordinate systems (see Section 3.2.1). The near and far plane of the viewing frustrum are set automatically, such that they encompass the bounding box of the 3D model. Then, the 3D model is rendered with the specified projection and modelview matrix.

In a next step, the depth buffer values that were written into the depth buffer during the rendering step are acquired from the graphics card. Then, the raw depth buffer values (which encode depth in relation the clipping planes) are converted back to depth values in the camera coordinate system. After this conversion, the 3D differences between the 3D model and the real measurements can efficiently be calculated pixelwise, by comparing the depth value of the synthetic depth image and the depth measurement of the depth camera at the same pixel.

Execution time For real-time 3D difference detection with a moving depth camera, the efficiency of the 3D difference detection is essential in order to fulfill the real-time constraint. Therefore, the execution time of the proposed approach is provided by Table 3.2.

Table 3.2 shows the sum of the execution times of the following algorithmic components of the 3D difference detection: the 3D mapping algorithm described in this subsection, as well as the 3D difference calculation and the 3D difference visualization (both described in more detail in Section 3.2.6 and implemented as a GPU fragment shader). The execution time was measured with an Intel Core i7 processor with 3.07 Ghz and an NVidia GeForce GTX 470. The CPU-based part of the framework is implemented as a single core C++ implementation.

If the 3D model consists of relatively few triangles, the execution time mainly results from the time it takes to copy the data to the graphics card and back to the CPU. This time is constant for a given image size. Even for large 3D models with more than two million triangles, the 3D difference detection takes only a few milliseconds and is thus real-time capable.

Please note that the values from Table 3.2 do not only include the time for the 3D mapping via analysis-by-synthesis, but also the execution time of the difference calculation and the visualization. The timing evaluation of the visualization includes an optional mapping of the detected differences onto a 2D image captured by an additional color camera (see Section 3.2.6 and publication [Kah13]). If the 3D differences are visualized by a direct color encoding (as in the figures shown in this thesis), each timing value specified for the 480×640 depth image decreases by 3-4 milliseconds.

Number of triangles	**Image size**		
	176×144	240×320	480×640
1.280	1ms	3ms	9ms
15.000	1ms	3ms	9ms
111.000	2ms	4ms	9ms
670.000	3ms	6ms	12ms
1.000.000	3ms	6ms	13ms
2.500.000	6ms	8ms	14ms

Table 3.2.: 3D difference detection: sum of the execution time of the 3D mapping, the difference calculation and the difference visualization (in milliseconds).

If the difference calculations and visualizations are implemented on the CPU, the additional execution time scales with the number of pixels the 3D model gets projected to. In contrast, due to the massive parallelisation of the calculations on the graphics card, this only has a very small effect on the GPU implementation. With a single-core implementation using the specified system hardware, a CPU implementation increases the specified calculation times for the 480×640 depth image by about 3 milliseconds if the actual differences only need to be calculated and visualized for few pixels (i.e., if there are many pixels the 3D model does not get projected to). The required calculation time increases by about 6-7 milliseconds if 3D differences are calculated and visualized for each pixel.

3.2.4. Depth image adjustment

Depth images are deformed by radial and tangential distortion effects. The perspective camera model introduced in Section 2.1.1 models the perspective projection of depth cameras. In order to account for distortion effects, each captured depth image is undistorted before it is used for the 3D difference detection.

While image based camera calibration procedures estimate the intrinsic parameters of the depth camera (focal length, principal point, skew and the distortion parameters), 3D calibration methods target the correction of systematic depth measurement errors. If a 3D calibration has been conducted, the depth values can be adjusted by applying the results of the 3D calibration process on the captured depth images. Approaches for the 3D calibration of depth cameras will be described in Section 5.1.1.

3.2.5. 3D reconstruction

The depth measurements acquired by depth cameras are affected by random noise as well as by systematic measurement errors (see Section 2.2.3). Furthermore, there can be gaps in the depth images at regions of pixel where no depth measurements could be acquired.

In order to address both measurement inaccuracies and missing 3D information, the 3D difference detection is complemented by a 3D reconstruction algorithm that reconstructs a 3D model from the captured depth images. Thus, gaps are filled from depth information acquired in other depth images. In addition, the fusion of 3D information from several depth images reduces the measurement noise. The surface of the captured scene is reconstructed in real time, while the depth camera is moved in order to detect 3D differences. Such a real-time 3D reconstruction algorithm, which reduces the measurement noise and the gaps, is detailed in Section 5.2.

Raw and reconstructed depth image In order to compare the reconstructed 3D model with the reference 3D model in real time, a depth image is extracted from the 3D model. While the depth image acquired by the depth camera is refered to by the term "raw depth image", the term "reconstructed depth image" refers to the depth image extracted from the reconstructed 3D model. If 3D reconstruction is applied, the reconstructed depth image replaces the raw depth image for the 3D difference calculation.

Extraction of the reconstructed depth image The reconstructed depth image can either be extracted from the 3D model with the analysis-by-synthesis approach described in Section 3.2.3, or by taking into account available information about the reconstructed 3D model. For example, the algorithm described in Section 5.2 estimates the reconstructed object surface with an implicit function. Within a fixed voxel grid, a signed distance function is estimated for each voxel. Thus, each voxel stores the distance of the voxel center to the closest point on a surface. For such an implicit surface representation within a voxel grid, the reconstructed depth image can be extracted by ray casting. The extraction of the depth image from the implicit surface via raycasting is described in Section 5.2.

3.2.6. Difference calculation and visualization

The depth image captured with the depth camera and the synthesized depth image are compared on a per-pixel basis. For each pixel of the depth image acquired by the depth camera, the depth value d_m of this pixel is compared with the depth value d_s of the corresponding pixel of the synthesized depth image. Here, the term "corresponding" refers to the pixel which has the same image coordinates.

Distance Metric The generalized mathematical representation of the Manhattan distance, the Euclidean distance and the Maximum distance is the Minkowski distance (also called p-norm). The definition of the Minkowski distance between two n-dimensional points x and y in the Euclidean space is provided by equation 3.1, with $p \geq 1$. For $p = 1$, the Minkowski distance is the Manhattan distance and for $p = 2$, the Minkowski distance equals the Euclidean distance.

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{3.1}$$

Depth images store a one-dimensional depth value per pixel. Thus, the distance between the depth values of two depth images at a certain pixel is the distance between two one-dimensional depth values. For $n = 1$, the Minkowski distance between the two depth values d_m and d_s is

$$|d_m - d_s| \tag{3.2}$$

for any $p \geq 1$. Thus, for $n = 1$, the Manhatten distance, the Euclidean distance and all other distance metrics represented by the Minkowski distance are the same.

As pointed out in Section 2.1.3, there are two different ways to encode depth in a depth image. If the stored depth represents a perspective projection, the stored depth is the Euclidean distance between the 3D point (in the camera coordinate system) and the optical camera center. In this case, the difference between the 1D depth values calculated with Equation 3.2 equals the Euclidean distance between 3D points in the camera coordinate system (the depth values can be converted to the 3D points with Equation 2.11).

On the other hand, if the depth represents an orthogonal projection, the depth value is the orthogonal distance between the 3D point in the camera coordinate system and the plane parallel to the viewing plane which intersects the optical camera center. In this case, the stored depth is the z-value of a 3D point (x, y, z) in the camera coordinate system. Thus, the difference between the 1D depth values calculated with Equation 3.2 is equivalent to the distance between the z-values of the 3D points. Again, this is a 1D distance calculation for which all norms that derive from the Minkowski distance in Equation 3.1 provide the same distance equation.

Difference visualization The 3D differences were calculated for each depth measurement and thus for each pixel of the depth image. Thus, the detected differences can be visualized pixelwise, by color

encoding. As a color encoding example, the difference visualizations shown in the figures of this thesis use the following colors (other encodings might be used as well):

Green	The real object matches the 3D model well (difference below a threshold).
Yellow	The real object is farther away than the virtual 3D model.
Red	The real object is closer than the virtual 3D model.

Blue	The view ray of this pixel does not intersect the 3D model (no 3D model at this pixel).
Black	At this pixel, no 3D measurement could be acquired by the depth camera.

Table 3.3.: Color encoding example: difference visualization used in this thesis.

In the difference visualizations shown in this thesis, the calculated differences are visualized as specified in Table 3.3.

Augmentation of captured images with difference visualization A slightly different visualization variant is an augmentation of the intensity (grey) images captured by the depth camera with a semi-transparent visualization of the calculated differences.

Similarly, the 2D image of a color camera rigidly coupled with the depth camera can be augmented with the difference visualization. If color images are augmented, the differences are not well visible because the colors of the 2D camera interfere with the colors of the difference visualization. Therefore, the color image first needs to be converted to a grayscale image, which is then augmented with the differences. In order to project the difference visualization onto the 2D image of the color camera, the depth values are first transformed to 3D points as described in Section 2.1.3. Based on the pose (R_D, t_D) of the depth camera, each 3D point p_{CCS} is then transformed from the camera coordinate system of the depth camera to the world coordinate system:

$$p_{WCS} = (R_D)^{-1} \cdot (p_{CCS} - t_D) \tag{3.3}$$

Finally, each 3D point p_{WCS} is projected from the world coordinate system to the 2D image coordinate system of the color camera with Equation 2.4 and the difference visualization is interpolated between the projected points.

3.3. Instantiations

This section introduces three different concrete instantiations of the general approach:

1. First, a basic approach is described which estimates the pose of the depth camera based on the intensity image of the depth camera itself. This basic approach uses the raw depth image for the 3D difference detection and does not include a 3D reconstruction step. This basic approach is easier to implement and to set up than the other two described concrete instantiations. However, it is also less accurate than the other two approaches.
2. A second concrete instantiation of the general concept uses a color camera rigidly coupled with the depth camera for image based camera pose estimation, in combination with 3D reconstruction for enhancing the accuracy of the 3D measurements. This approach provides intermediate accuracy.
3. The third concrete instantiation estimates the pose of the depth camera with a precise coordinate measuring machine. Furthermore, the accuracy of the 3D difference detection is enhanced with 3D reconstruction. This instantiation provides the highest accuracy and is thus the proposed approach for precise, real time 3D difference detection with a moving depth camera.

3.3.1. Basic approach (without tracking device and without 3D reconstruction)

Figure 3.3 sketches the algorithmic components and the data flow of a simple approach for real-time 3D difference detection. Here, no additional tracking device is used. Instead, the pose of the depth camera is estimated by detecting a square marker in the intensity image of the depth camera. Furthermore, the raw depth image as captured by the depth image is used for the 3D difference detection.

Figure 3.3 shows an early, basic demonstrator for real-time 3D difference detection with a hand-held depth camera. The coordinates of the square marker attached to the brick model (Figure 3.3a) are specified in the coordinate system of the virtual 3D model (Figure 3.3b). Thus, in this setup, the tracking coordinate system is identical to the coordinate system of the 3D model. This provides an implicit alignment of both coordinate systems.

By detecting the image marker in the 2D camera image and estimating the camera pose with image-based camera tracking, the position and orientation of the depth camera is calculated relative to the 3D model's coordinate system. Then, the 3D model is rendered from the estimated pose of the depth camera to acquire an artifical depth image from the depth buffer of the graphics card. Each such synthesized depth value is compared to the real depth measurement acquired by the depth camera.

Figure 3.3c shows a rotated view of the 3D point cloud acquired with a time-of-flight depth camera. In Figure 3.3d, the detected differences are visualized with a color based augmentation of the depth camera's intensity image. The plate of the 3D model shown in Figure 3.3 has a size of 38×38 cm. The distance of the time-of-flight depth camera to the 3D model is about 50-90cm. In this figure, differences of more than 5cm are visualized by the color encoding.

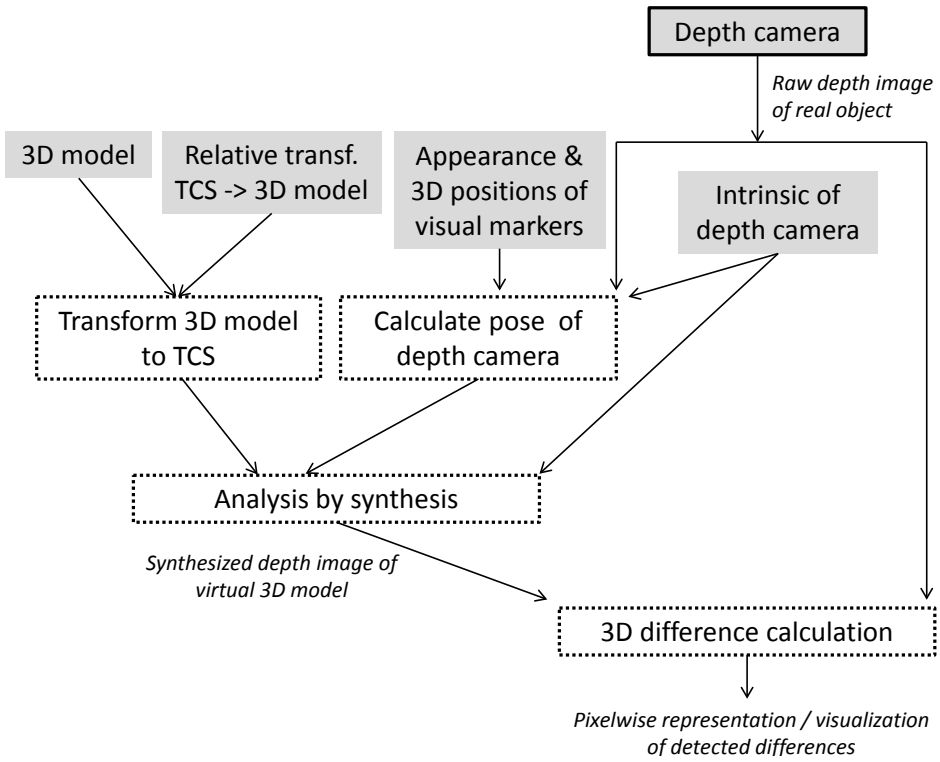
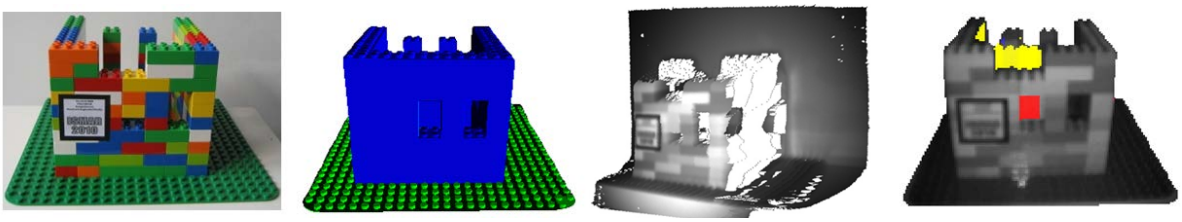


Figure 3.2.: Basic approach for 3D difference detection: without additional tracking device and without 3D reconstruction.



(a) Real object (b) Virtual 3D model (c) 3D point cloud (d) Visualized 3D differences

Figure 3.3.: First demonstrator of basic approach for 3D difference detection.

3.3.2. 2D image based camera pose estimation (reconstructed feature map)

Figure 3.4 illustrates the main algorithmic components and the data flow of a 3D difference detection in which an additional tracking device is used for the estimation of the depth camera's pose. In this instantiation, the additional tracking device is a color camera which is rigidly coupled with the depth camera. Figure 3.5 shows real-time 3D difference detection with a hand-held Kinect. This device contains both a color camera and a depth camera (see Section 2.2.2).

In the setup of Figure 3.5 and for the difference detection results shown in Figure 3.6, the pose of the color camera is estimated with a reconstructed 3D feature map [WWK11]. This 3D feature map is reconstructed and aligned with the 3D model coordinate system in an offline preparation step, in order to use the reconstructed 3D features for the image based camera pose estimation during runtime of the 3D difference detection.

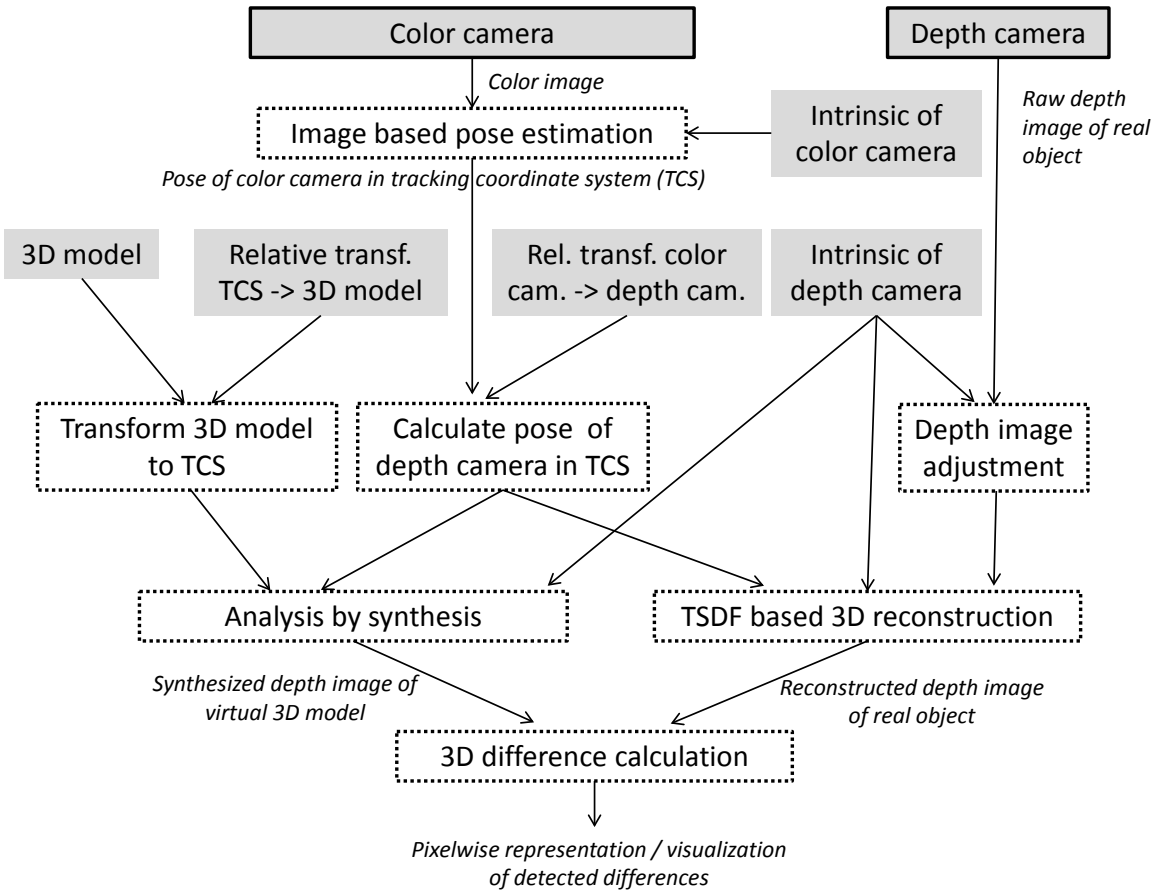


Figure 3.4.: 3D difference detection: image based camera pose estimation using a reconstructed feature map. With 3D reconstruction based on captured depth measurements.

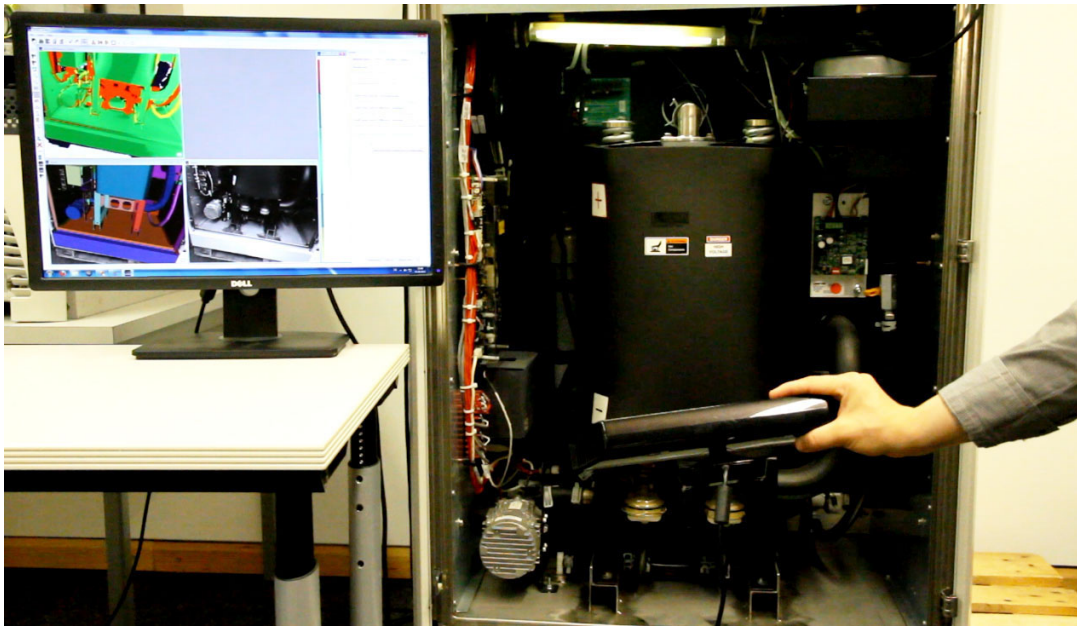


Figure 3.5.: 3D difference detection with a hand-held Kinect.

Offline preparation The intrinsic parameters of both the color camera and the depth camera are required for the image based camera pose estimation and the 3D difference detection. Thus, both cameras need to be calibrated intrinsically if their internal projection parameters were not estimated yet. Furthermore, if the relative transformation between the color camera and the depth camera is not known yet, it needs to be estimated as part of the offline calibration. Just as for a stereo calibration between two rigidly coupled color cameras, the relative transformation between a color and a depth cameras is calculated by detecting a 2D calibration pattern with known size and appearance in the images of both cameras [SBK08] [SJP13].

In order to reconstruct a 3D feature map for the image based camera pose estimation, first a 2D image sequence of the scene is recorded with the 2D color camera which is used as tracking device. Based on this 2D image sequence captured from various viewpoints, the 3D positions of tracked 2D image features are reconstructed by 3D triangulation as part of a simultaneous localization and mapping approach [WWK11]. In order to align the reconstructed feature map with the 3D model, the user needs to select a number of reconstructed 3D points and their corresponding positions on the surface of the 3D model. The rigid transformation between both coordinate systems is calculated based on these 3D-3D correspondences: first, the centroids of both 3D point sets are aligned. Then, a rotation is estimated with singular value decomposition which minimizes the distances between the 3D points of the two point sets [Ume91].

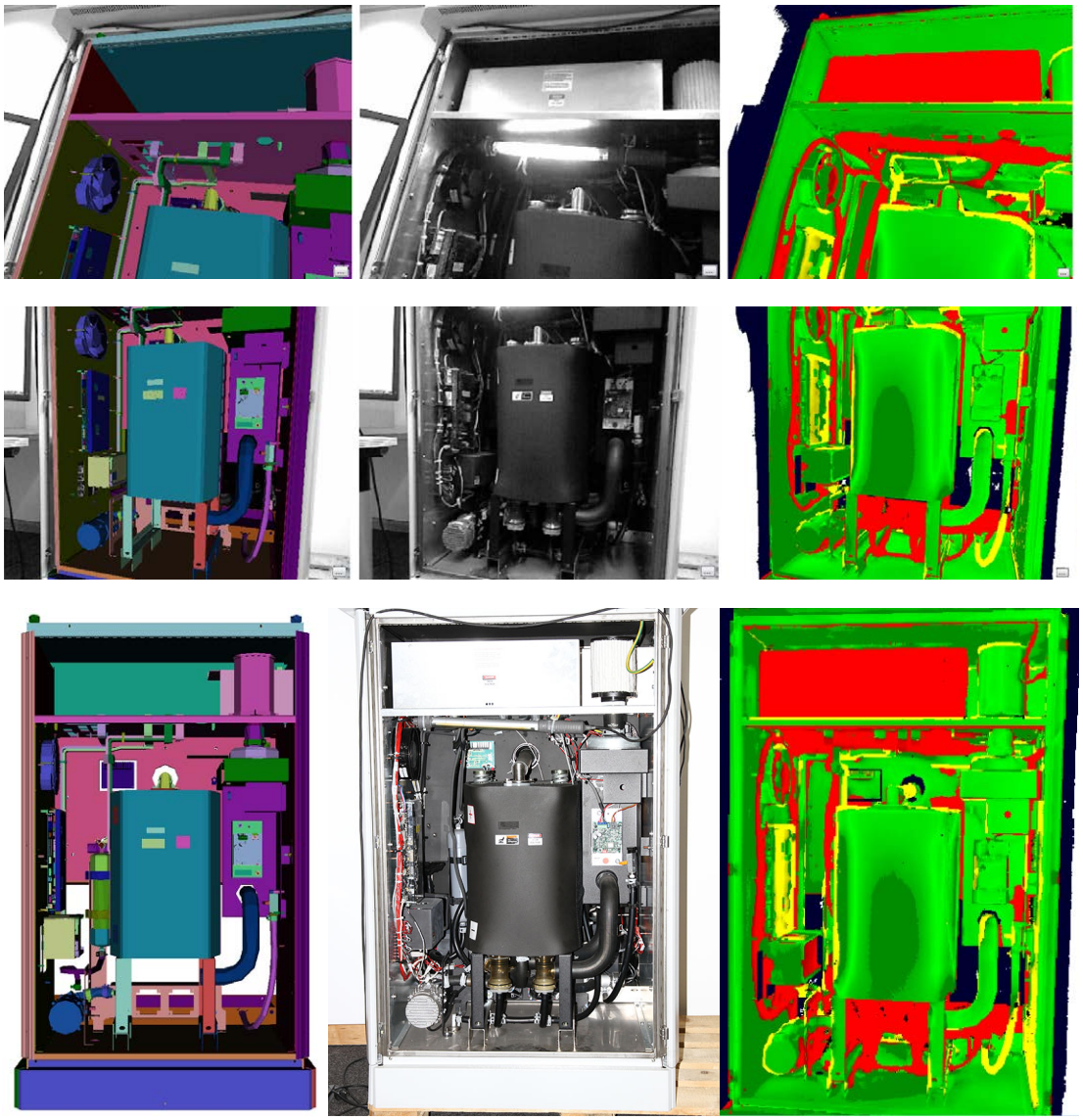
In a next step, the 3D reconstruction is refined with a contrained bundle adjustment. Furthermore, the visibility of each feature from different viewpoints is analyzed and a randomized trees classifier is trained based on the captured 2D image sequence [LF06]. This classifier provides a set of reconstructed 3D features and the appearances of these features in 2D images captured from different viewpoints which can be used for a wide-baseline initialization of the camera pose estimation [WWK11].

Camera pose estimation To initialize the 3D difference detection during runtime, the pose of the 2D camera is first initialized with randomized trees, based on the reconstructed 3D feature map. While the 2D camera is moved, new 2D features are extracted and reconstructed by triangulation. The pose of the 2D camera is estimated both from the 3D points reconstructed during runtime and from the 3D feature map reconstructed in the offline preparation [BWS06] [WWK11]. For each captured frame, the pose of the depth camera is estimated from the pose of the color camera and the relative transformation $(\Delta R, \Delta t)$ between both cameras as described in Section 3.2.2.

3D reconstruction based on depth images As described in Section 3.2.5, during runtime, 3D reconstruction can be used to fill gaps and to reduce the inaccuracies of 3D measurements acquired by depth cameras. In contrast to the 3D reconstruction from 2D images used for the image based camera pose estimation, this 3D reconstruction based on the captured depth images is a dense reconstruction. This dense 3D reconstruction is applied in order to reduce the measurement inaccuracies of the depth camera. Thus, for the 3D difference detection, the raw depth image acquired by the depth camera is replaced by a reconstructed depth image that is extracted from the 3D reconstruction. The 3D reconstruction will be described in more detail in Section 5.2.

Execution time The runtime of this setup was evaluated with an Intel Core i7 processor with 3.07 Ghz and an NVidia GeForce GTX 470 (with the same system as used for the timing provided in Section 3.2.3). Here, the 3D difference detection as well as the camera pose estimation were implemented as a single core C++ implementation on a CPU. In contrast, the 3D reconstruction algorithm was a CUDA implementation on the GPU.

For a 3D model of the fuel cell that consists of 407.000 triangles, the execution time of this setup that uses 2D image based camera pose estimation is about 143 milliseconds. This corresponds to a framerate of 7 frames per second. In more detail, the processing time is 13 ms for the 3D difference detection, 40 ms for the 3D reconstruction algorithm and about 90ms for the image preprocessing and the image based camera pose estimation. The time required to calculate the pose of the depth camera from the estimated pose of the color camera is well below 1 ms. Thus, with this approach, most of the processing time is required for the image based pose estimation of the 2D color camera which is used as tracking device. This is much more computationally expensive than the pose estimation with a coordinate measuring machine, which is next described as the proposed approach for precise real-time 3D difference detection.



(c) Virtual 3D model. (d) Real object (fuel cell). (e) Difference visualization (based on the 3D reconstruction).

Figure 3.6.: 3D difference detection of a fuel cell, acquired with the instantiation illustrated in Figure 3.4 and the setup shown in Figure 3.5. Color encoding of the measured differences: bright green: < 20 mm. Dark green: < 40 mm. Red and yellow: > 40 mm.

3.3.3. Pose estimation with a coordinate measuring machine (measurement arm)

Figure 3.7 provides an overview of 3D difference detection with a high precision. This instantiation of the general 3D difference detection concept includes two main approaches for enhancing the accuracy of the 3D difference detection. On the one hand, the depth camera is combined with a coordinate measuring machine which provides a precise pose estimation. On the other hand, dense real-time 3D reconstruction is used to improve the accuracy of the depth measurements acquired by the depth camera and to fill missing gaps in the depth image. Figure 3.8 shows a setup for precise 3D difference detection which uses a Kinect depth camera rigidly coupled with a measurement arm, in combination with dense 3D reconstruction. The precise pose estimation with the measurement arm is described in detail in Chapter 4 and the 3D reconstruction in Chapter 5.

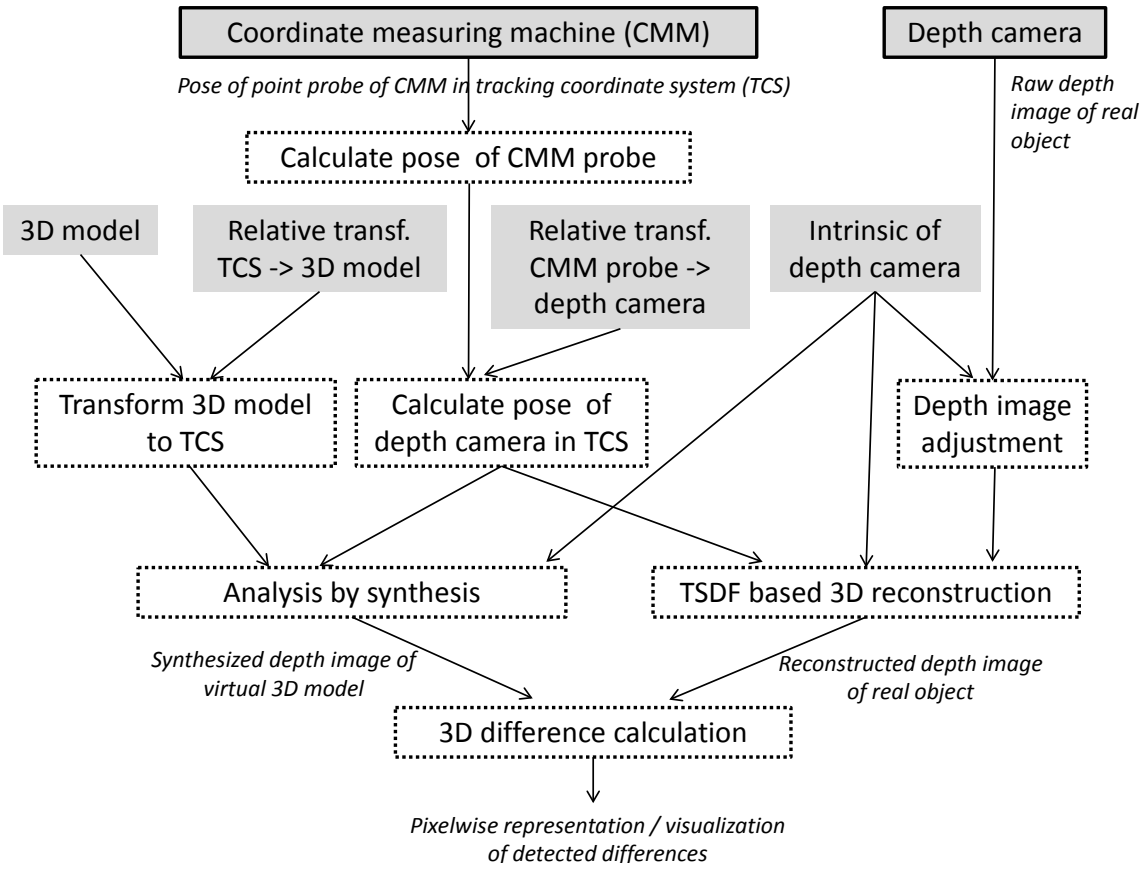


Figure 3.7.: Precise 3D difference detection: pose estimation with a coordinate measuring machine (CMM). In addition, dense real-time 3D reconstruction enhances the accuracy of depth measurements.

Execution time With the same hardware as used for the runtime evaluation of Section 3.3.2 (single core implementation on a 3.07 Ghz Core i7 and an NVidia GeForce GTX 470 graphics card), the time required for estimating the pose of the depth camera is shorter than 1 millisecond. This time includes both the acquisition of the pose of the measurement arm (R_1, t_1) and the calculation of the depth camera's pose (R_2, t_2) from (R_1, t_1) using the relative transformation $(\Delta R, \Delta t)$ between both devices. For a 3D model with 407.000 triangles, the processing time is 13 ms for the 3D difference detection (CPU implementation) and 40 ms for the 3D reconstruction algorithm (on a GPU). The overall framerate is about 16-17 frames per second. As the pose estimation takes less than 1 ms, this approach has a higher framerate than approaches which use a 2D color camera as tracking device. A further comparison of both approaches is provided in Section 4.1 and a quantitative accuracy evaluation in Chapter 6.

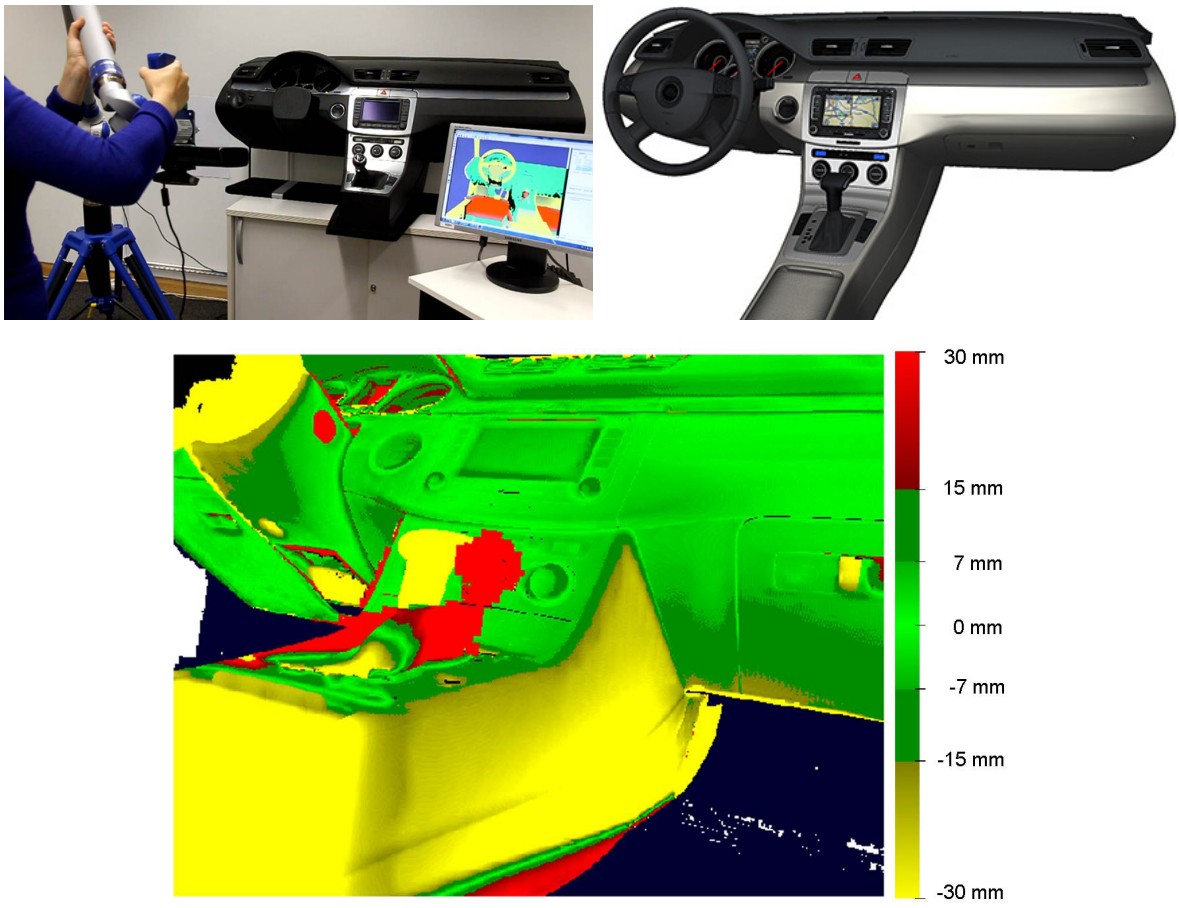


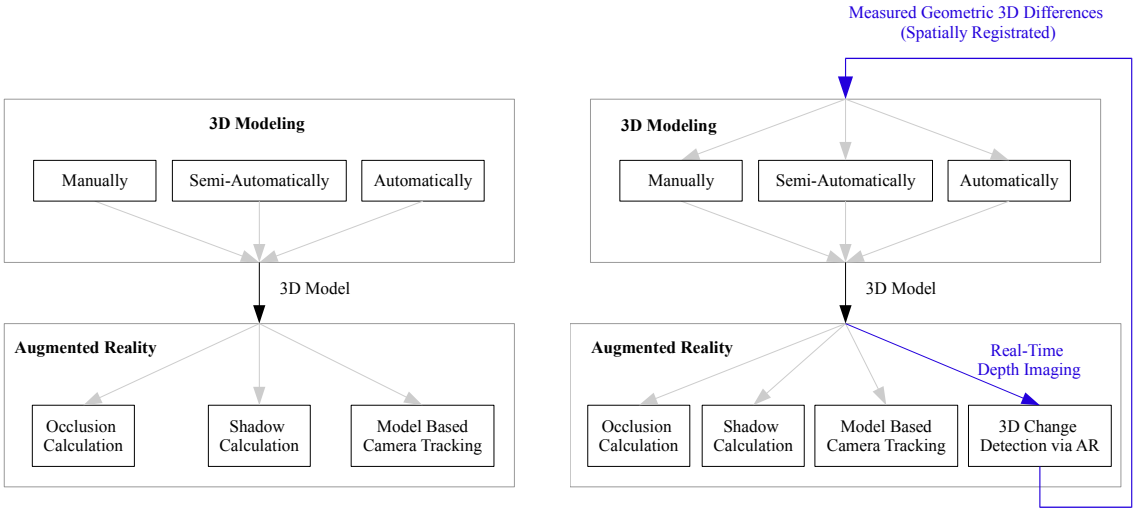
Figure 3.8.: 3D difference detection with a Kinect depth camera, pose estimation by a coordinate measuring machine (a measurement arm) and 3D reconstruction.

3.4. Closing the loop between 3D modeling and augmented reality

While the next chapters or this thesis will focus on 3D difference detection with a high precision, this section briefly sketches how real-time 3D difference detection can support 3D modeling and augmented reality applications [Kah13]. Augmented reality (AR) applications combine real and virtual, are interactive in real time and registered in 3D [Azu97]. Since Azuma first stated these characteristics of AR applications in 1997, augmented reality has matured remarkably [ZDB08]. However, an important bottleneck remains: The availability of 3D models of real scenes, which correctly model not only sparse point but also the surface of the scene. Such dense 3D models are important for two different augmented reality aspects. First, a 3D model is needed for a smooth and seamless integration of virtual objects into the camera images. Therefore, the virtual objects should be illuminated in a consistent way with the illumination of the real scene, they should cast shadows and they should be occluded by parts of the real scene which are closer than the virtual object. Both occlusion handling and shadow calculation require knowledge about the 3D structure of the real scene [Hal04] [PSP09]. Furthermore, dense 3D models are often used for model-based camera tracking, both for the camera pose initialisation and for model-based frame-to-frame tracking [LF05]. Model-based estimations of the camera pose have the advantage that they overcome the need to prepare the scene with fiducial markers [GRS06].

Why a dense 3D model is required even though depth cameras capture dense 3D measurements in real time Depth cameras can be used to create realistic AR applications with shadow mapping or occlusion culling even if no 3D model of the real scene exists [FKOJ11]. However, these approaches have several drawbacks. First, the user can only use the AR application with a depth camera. While depth cameras have become widely distributed for desktop computers, they are not integrated into most mobile devices. A 3D model offers better device independence: It can also be used by such mobile devices which offer powerful processors and built-in 2D cameras and which are thus suited for model-based AR, but which have no integrated depth measurement devices. Another disadvantage of real-time depth imaging without a 3D model is that the 2D camera and the depth camera capture the scene from different viewpoints. Thus, due to occlusions, there is no complete mapping between the color pixels and the depth measurements [LKH07]. Further artifacts arise in fast camera movements if the color and the depth camera capture the images at slightly different points in time. Finally, a 3D model can provide more stable 2D-3D correspondences for model-based camera tracking than depth measurements which are captured on the fly. For example, depth measurement artifacts occur due to motion blur effects if the depth camera is moved quickly. In the 3D model creation process, these artifacts can easily be circumvented by slow camera movements. Therefore, if the user moves the camera quickly during the tracking, acquiring the 3D measurements from the depth camera on the fly would suffer from these artifacts whereas the 3D model provides 3D surface information which is not influenced by this effect.

Acquisition of a dense 3D model Many AR applications which use dense 3D models implicitly assume that such a 3D model already exists. Therefore, the reconstruction process is often decoupled from the AR application and the reconstruction is part of an offline preparation phase before the AR



(a) Conventional approach

(b) With 3D difference detection

Figure 3.9.: (a) Conventional approach: The 3D model is used as input for the AR application, but not vice versa. (b) Proposed approach: 3D differences are detected and fed back into the 3D modeling pipeline, in order to update the 3D model.

application can be used. This strict separation between the modeling process and the application of the created 3D model for AR causes two major problems: First, the occurence of a change is often not obvious in the first place. Furthermore, even when the user is aware that the 3D model does not fit the reality any more, it is often a difficult task to find out how the 3D model needs to be adapted such that it correctly models the real scene again.

Supporting 3D modeling for AR with 3D difference detection While the existing 3D model could be discarded and replaced by a completely new 3D reconstruction, it is often advantageous to keep those parts of the 3D model which have not changed (for example because they are modeled efficiently with few triangles, or in order to keep internal structures of the 3D model). This task can be solved with 3D difference detection.

In order to use 3D difference detection for 3D modeling, in a first step, 3D differences can be detected as described in this thesis. In a next step, the measured 3D differences can be fed back into the 3D modeling pipeline where they can be used to update the 3D model either manually or semi-automatically [Kah13]. This update step benefits from the fact that the 3D difference detection registers the measured depth values and the 3D model in a common coordinate system. This eases the model adjustment task, both for the user and for (semi-)automatic model adjustment.

3.5. Conclusion

This chapter introduced depth image based, real-time 3D difference detection with a hand-held, moving depth camera. Previous approaches for difference detection were either restricted to a static camera position, not suited for arbitrary 3D models or required the manual specification of 3D correspondences between a laser scan of a construction site and a 3D model of the construction site for each new scan position, in order to transform both data sets into a common coordinate system. In contrast, the approach for depth camera based difference detection described in this chapter introduced a solution for real-time 3D difference detection with a moving camera. Thus, it provided an answer to the question **Q1: "How can 3D differences be detected in real time and from arbitrary viewpoints using a single depth camera?"**

The presented approach for real-time 3D difference detection with a depth camera is based on the fusion of computer vision and computer graphics. This chapter described the integration of computer vision methods (depth image acquisition and processing as well as pose estimation) with a computer graphics based analysis-by-synthesis approach.

The proposed approach can be used to efficiently compare a depth image with a 3D model in real time, with an update rate of several frames per second. For example, all differences between a depth image that contains 307.200 depth measurements and a 3D model with 2.5 million triangles can be calculated and visualized in less than 15 milliseconds. Furthermore, the internal structure of the 3D model does not need to be parsed and the proposed approach is applicable for any 3D model which can be rendered, irrespective of the internal representation of the 3D data. Thus, this chapter provided an answer to question **Q1.1: "How can the 3D measurements of a depth camera be mapped onto an arbitrary 3D model in real time?"**

This chapter described the general concept for real-time 3D difference detection, the main algorithmic components and concrete instantiations of the general concept. These concrete examples illustrate how the proposed approach can be used to detect differences with a moving depth camera. Three concrete examples of the general difference detection concept were illustrated. First, a basic setup was described which uses only the depth camera, no additional tracking device, and which detects differences based on the raw captured depth measurements. A second concrete instantiation uses an additional color camera as tracking device and estimates the pose of the camera with image based camera pose estimation. Furthermore, a dense 3D model is reconstructed during the 3D difference detection and the raw depth measurements acquired by the depth camera are replaced by the 3D reconstruction. The third instantiation is the proposed approach for precise 3D difference detection with a moving depth camera. This approach also enhances the measurement accuracy with 3D reconstruction. Furthermore, it uses a coordinate measuring machine as tracking device which provides a precise pose estimation.

By the description of the abstract, main algorithmic components in Section 3.2 and the illustration of conrete instantiations in Section 3.3, this chapter provided an answer to the question **Q1.2: "Given a mapping of 3D measurements onto a 3D model, how can 3D differences be detected in real time for a moving depth camera?"**

Sources of inaccuracies While this chapter introduced the general concept for real-time 3D difference detection, the next two chapters will address the question how 3D differences can be detected with a high accuracy. Therefore, the sources of inaccuracies that affect the accuracy of the 3D difference detection are listed in this section. Then, solutions for reducing these inaccuracies are discussed and detailed in the next two chapters. The inaccuracies that arise in the 3D difference detection can be grouped in two categories:

1. Inaccuracies in the depth camera's pose estimation
   - Inaccuracies of the pose estimation device
   - Inaccurate relative transformation between the pose estimation device and the depth camera
   - Inaccurate alignment of the tracking coordinate system and the 3D model coordinate system
   - Temporal offset between the pose acquisition by the pose estimation device and the depth image acquisition by the depth camera
2. Measurement inaccuracies of the depth camera
   - Random measurement noise
   - Systematic measurement errors
   - Motion blur effects

Inaccuracies of the depth camera's pose estimation Inaccuracies of the pose estimation reduce the accuracy of the difference detection. If an additional pose estimation device is used, the overall accuracy also depends on the accuracy of the estimated relative transformation between the depth camera and the pose estimation device. Furthermore, the accuracy decreases if the 3D model coordinate system is not accurately aligned with the tracking coordinate system (in which the pose of the camera is estimated). Finally, the accuracy also decreases if the depth image was acquired at another timestamp than the time at which the pose was estimated. These issues will be addressed in Chapter 4.

Measurement inaccuracies of the depth camera As described in Section 2.2.3, depth cameras suffer both from random measurement noise and from systematic measurement errors. Furthermore, due to motion blur effects, the accuracy decreases if either the depth camera or the captured objects move with high speed. As the inspection of differences is more difficult during fast than during slow camera movements anyway (independent of motion blur effects), motion blur is less important for 3D difference detection than random measurement noise and systematic measurement errors. The reduction of measurement inaccuracies of the depth camera is addressed in Chapter 5.

4. Precise pose estimation

For accurate 3D difference detection, the pose of the depth camera relative to the 3D model needs to be estimated with a high precision. Therefore, this chapter addresses the question Q2.1:

Q2.1 How can precise pose estimation be integrated in the 3D difference detection?

First, Section 4.1 describes and discusses possible approaches to estimate the position and orientation of a depth camera relative to a 3D model. Then, pose estimation with a coordinate measuring machine is detailed in Section 4.2, with a focus on the estimation of the relative transformation between the depth camera and the coordinate measuring machine. Therefore, Section 4.2 introduces and comparatively evaluates both a 2D image based and a 3D measurement based approach for the hand-eye calibration between a depth camera and a coordinate measuring machine. It is based on the publications [KK12] and [KHW14].

4.1. Discussion of approaches

This section describes and discusses three possible approaches to estimate the position and orientation of a depth camera relative to a 3D model: image based camera tracking, geometric registration of the depth measurements and pose estimation with a coordinate measuring machine.

4.1.1. Image based camera pose estimation

Image based camera pose estimation analyzes the 2D image acquired by a depth camera or an additional 2D camera, in order to estimate the pose of the camera relative to a fixed tracking coordinate system. In the context of depth image based 3D difference detection, the following image sources can be used for image based camera pose estimation:

- Infrared / intensity image acquired by a depth camera, e.g.
  - intensity images acquired by a time-of-flight depth camera or
  - infrared images acquired by a structured light depth camera
- Image of a second camera (e.g. a color camera), either
  - rigidly coupled with the depth camera (inside-out tracking) or
  - a camera which tracks a marker rigidly coupled with the depth camera (outside-in tracking)

The intensity images acquired by state-of-the art depth cameras have a low resolution ($176 \cdot 144$ respectively $204 \cdot 204$ pixels) and the Kinect structured light camera can not output infrared images while providing depth information. Instead, a color camera (such as the RGB camera which is part of the Kinect) can be used for image based pose estimation. Thus, for image based camera pose estimation, the tracking device is either the depth camera itself or an additional camera. If an additional color camera is used, inaccuracies in the estimation of the relative transformation between the color camera and the depth camera reduce the overall accuracy of the 3D difference detection. However, accurate stereo calibration algorithms have been proposed for estimating the relative transformation between two cameras, e.g. by Tsai [Tsa87] and by Zhang [Zha99].

A large number of different real-time approaches for image based camera pose estimation have been proposed. For example, these include marker based camera pose estimation [PMK06], model based camera pose estimation [Wue08] or point based camera pose estimation [ST94] [BBS07] [WRM*08] [WWK11]. Point based camera pose estimation algorithms detect characteristic point features in the 2D camera images. With structure from motion respectively simultaneous localization and mapping (SLAM), the 3D positions are reconstructed from a 2D image sequence and the pose of the depth camera is estimated at the same time, for each frame.

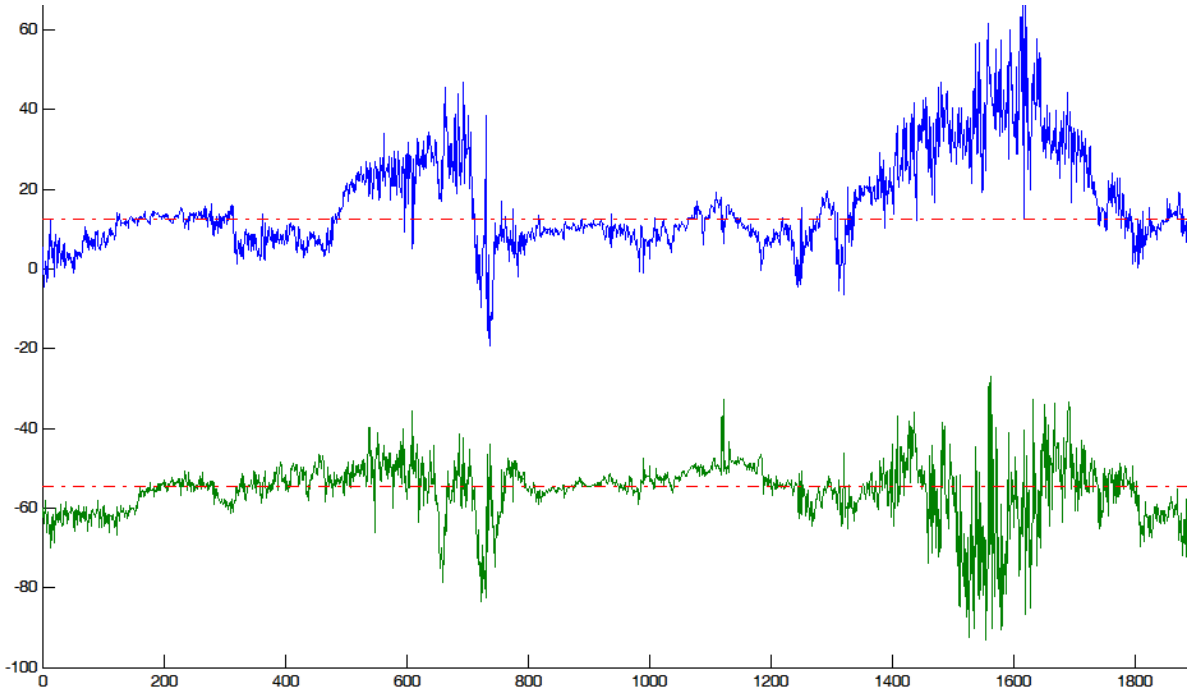


Figure 4.1.: Variation of image based pose estimation (four 2D-3D correspondences, image marker tracker) in mm. Blue: t1, green: t2 of translation t=(t1, t2, t3). Red: median of t1 and t2.

Accuracy The accuracy of image based camera tracking depends on the captured scene: The camera pose can only be estimated with an image based approach if enough stable, characteristic image features are visible in the captured camera image. Furthermore, the accuracy of image based camera tracking is sensitive to the distribution of the detected features in the 2D camera image. The accuracy decreases if the tracked features are not evenly distributed in the whole camera image (for example, if features can only be found in a part of the camera image).

With image based camera pose estimation, the estimated camera pose can differ from the real camera pose by several millimeters to centimeters. Figure 4.1 shows the variations of the parameter t of a pose (R,t) estimated with an image marker.

First, the image marker was detected in each of 1900 two dimensional infrared images, captured by a moving Kinect depth camera. Then, for each frame, the pose (R_D,t_D) of the depth camera relative to the marker was estimated with image based camera pose estimation. The pose was estimated from the four 2D-3D correspondences of the marker corners and their projections on the infrared images. As the depth camera has a different pose in each frame, the poses cannot be directly compared to each other in order to analyze variations. Therefore, each pose (R_D,t_D) was compared to the pose (R_T,t_T) of a tracking device (a measurement arm) rigidly coupled with the depth camera.

Thus, (R,t) is the relative transformation between the pose (R_D,t_D) of the depth camera and the pose (R_T,t_T). As the tracking device estimates its pose with a high precision (better than 0.1 mm), the relative transformation (R,t) would be constant if there were no errors in the image based camera pose estimation. Thus, in Figure 4.1, the variations of t are image based camera pose estimation errors. These errors vary from several millimeters to several centimeters.

Wientapper et al. evaluated the accuracy of feature based camera tracking [WWK11]. In this image based camera pose estimation approach, the camera pose is not estimated with a marker tracker, but by detecting characteristic 2D image features in subsequent camera images. By reconstructing the 3D positions of the tracked 2D image features via triangulation, the reconstructed 3D points and their detected 2D positions provided the 2D-3D correspondences for image based camera pose estimation. Thus, in contrast to marker based pose estimation, the camera pose was estimated from more than four 2D-3D correspondences. However, the evaluation results of the accuracy were similar as for the marker based pose estimation. For this structure from motion based pose estimation approach, the estimated camera position also differed several millimeters to centimeters from the reference positions of the camera.

4.1.2. Geometric 3D registration

As an alternative to 2D image based camera tracking, the depth measurements acquired by the depth camera could also be used to calculate the pose of the depth camera [Wun10] [NIH*11]. The pose could either be estimated by geometrically aligning the current depth image with the virtual 3D model or by aligning the current depth image with 3D measurements acquired from previous depth images. Therefore, the pose of the depth camera could be estimated by minimizing the geometric 3D distances between the depth image and the other 3D data with the Iterative Closest Point algorithm [BM92]

[RL01] [CSK05]. Geometric registration is computationally very expensive (especially, if a large number of 3D-3D correspondences are aligned to find a robust solution for the registration). Geometric registration on the CPU is not real-time capable for depth images. In contrast to CPU based geometric registration, Newcombe et al. [NIH*11] showed that a geometric real-time registration is feasible with a highly parallelized implementation on a graphics card. However, similar to image based camera tracking, the accuracy of camera pose estimation based on geometric features depends on the structure of the captured scene.

For geometric camera pose estimation, the captured scene needs to have a 3D shape for which the camera pose can be calculated unambiguously from the captured depth images. This condition is often not fulfilled. For example, if a depth camera captures several parallel pipes in front of a wall, the camera pose cannot be calculated unambiguously from the depth image as there is one degree of freedom along the pipes. If the captured depth image contains mainly planar shapes or if the user moves the camera close to the surface in order to inspect details, an unambiguous registration of the depth image often is difficult. In this case case, the registration diverges and the pose of the depth camera cannot be estimated.

4.1.3. Robots and coordinate measuring machines

In industrial applications, industrial robots [Nof99] [SKK*10] and coordinate measuring machines are commonly used for 3D measuring tasks which require a high precision [Tan92]. While robots are controlled by a computer program, measurement arms are typically operated hand-held. Measurement arms have a point tip with which 3D positions of points on the surface of an object can be measured. Thus, the arm is moved by a user which points the tip of the arm at a 3D position whose 3D coordinates should be measured. In the context of 3D difference detection, such 3D measurements on object surfaces can be used to align a 3D model of this object with the coordinate system of the measurement arm, respectively with 3D points on the surface of the real object (see Section 4.2.2).

The point tip of a measurement arm is attached to the base of the arm by several rigid elements, which are linked by rotational joints. A measurement arm measures the rotation of each joint (for example with shaft encoders) and outputs the position and orientation of its point tip relative to its base coordinate system. Measurement arms have a high update rate and typically provide more than 60 updates per second.

A measurement arm does not only provide the 3D position of its point tip, but also its orientation. Thus, it can be used as a tracking device which measures the pose (R,t) of its point tip. In previous publications by Gruber et al. [GGV*10] and Lieberknecht et al. [LBMN11], 2D color cameras were combined with a measurement arm in order to acquire ground truth data for the evaluation of image based camera pose estimation algorithms.

In contrast to industrial robots, measurement arms are often portable, such that they can be moved and positioned at a different location. Both industrial robots and measurement arms provide a high measurement accuracy. For example, a Faro Platinum measurement arm has a measurement range of 3.7 meters and provides a pose with a precision better than 0.073mm.

In contrast to both image based and geometric alignment based pose estimation, the accuracy of the pose estimation is independent of the captured scene. With a measurement arm, the camera pose estimation cannot fail due to ambiguities or due to too few optical or geometric features. Thus, if a measurement arm is used for the pose estimation, the user does not need to be careful when moving the camera, in order to avoid tracking loss. Instead, the user can concentrate on the 3D difference detection task at hand.

4.1.4. Discussion

Table 4.1 lists the main strengths and weaknesses of image based pose estimation, geometric pose estimation and pose estimation with a coordinate measuring machine.

Both image based and geometric approaches for camera pose estimation require characteristic features that can be detected in the captured images. While image based pose estimation approaches require 2D features, unambiguous 3D shapes are required for a geometric registration.

In contrast, pose estimation with a coordinate measuring machine (such as a measurement arm) has the advantage that the pose estimation does not depend on the structure of the captured scene. Thus, the pose estimation accuracy neither decreases due to too few detected features nor does the pose estimation need to be re-initialized. A coordinate measurement machine continously provides a pose estimation and the pose estimation cannot fail due to characteristics of the captured scene. In contrast, 2D image based and depth image based pose estimation fails if too few features could be detected.

While the position estimated with image based or with geometric pose estimation can differ from the real position by several millimeters to centimeters, a coordinate measuring machine such as a Faro Platinum measurement arm outputs a pose with an accuracy better than 0.1 mm. Furthermore, pose estimation with a coordinate measurement machine requires less computational ressources than image based or depth image based pose estimation. The pose of the point tip of a measurement arm is internally measured by the arm and directly output by this device, so the only calculation required at runtime is the calculation of the pose of the depth camera from the pose of the measurement arm. This can be calculated in less than one millisecond.

Coordinate measurement machines can measure sparse 3D points on the surface of objects with the same accuracy as the accuracy of their pose estimation. Such 3D measurements can be used to align the 3D model coordinate system with the tracking coordinate system. The accuracy of these measurements is higher than the accuracy of 3D positions estimated with image based 3D reconstruction or measured with a depth camera. Thus, the coordinate systems can be aligned more accurately with a coordinate measuring machine than with image based or depth image based approaches for pose estimation.

For these reasons, the proposed approach for 3D difference detection with high precision pose estimation is the combination of a depth camera with a coordinate measuring machine, such as a measurement arm. Therefore, pose estimation with a measurement arm will be described in the next section, with a focus on the estimation of the relative transformation between the depth camera and the measurement arm.

	Image based pose estimation	**Geometric pose estimation**	**Pose estimation with a CMM (measurement arm)**
Precision	Intermediate (depends on availability of characteristic 2D image features), estimated position can differ from ground truth by several mm to cm.	Intermediate (depends on 3D shape of captured depth image and the accuracy of the captured depth measurements), estimated position can differ from ground truth by several mm to cm.	Very precise (estimated position differs <0.1mm from ground truth).
Dependency on captured scene	Requires characteristic 2D image features (pose estimation accuracy decreases if too few characteristic image features can be tracked).	Requires characteristic 3D shapes (3D shape needs to be unambiguous in all dimensions).	None, does not require any information about the captured environment (CMM measures internal joint rotations).
Tracking loss, divergence of pose estimation	Pose estimation can fail (e.g. if the captured 2D image contains few characteristic 2D features).	Pose estimation can fail (e.g. if the captured 3D shape has a degree of freedom in one direction).	No tracking loss (pose estimation does not depend on captured image).
Re-initialization of pose estimation	Eventually required (depends on pose estimation approach).	Required.	Not required, pose available for every frame.
Computational complexity	Medium.	High (usually not real-time capable on a CPU, but with a GPU implementation).	Low, very fast (measured pose is directly provided by CMM device).
Camera movement	Hand-held, only camera(s).	Hand-held, only camera.	Camera coupled with the arm, thus both the arm and the camera are moved manually.
Additional financial costs	None.	None.	High (costs for CMM).

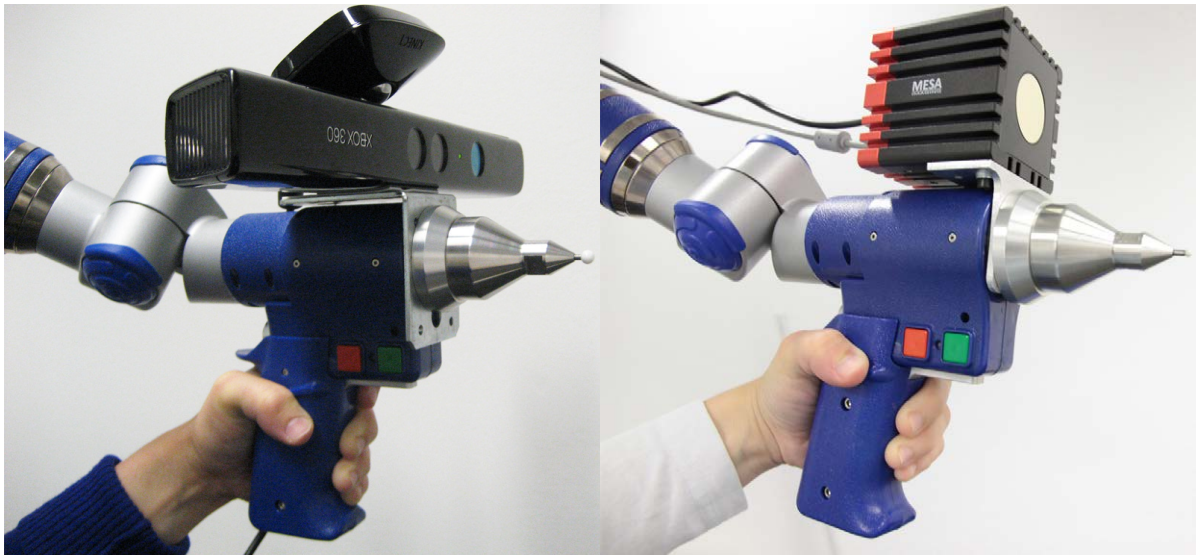
Table 4.1.: Comparison of image based pose estimation, geometric (depth measurement based) pose estimation and pose estimation with a coordinate measuring machine (a measurement arm).

Figure 4.2.: 3D difference detection with a Kinect depth camera and an industrial measurement arm.

4.2. Pose estimation with a coordinate measuring machine

Figure 4.2 shows a setup in which a depth camera rigidly coupled with a measurement arm is used to detect differences between a real object and a 3D model of this object. Here, the industrial measurement arm is used for the pose estimation of the depth camera. In order to track the pose of a depth camera with a measurement arm, the depth camera is rigidly coupled with the measurement arm. To transform the 3D measurements of the depth camera into the coordinate system of the articulated arm, the relative transformation between the depth camera and the measurement arm needs to be known. This transformation is called "hand-eye transformation". As depth cameras acquire both 2D images and 3D measurements, there are two different approaches for the hand-eye calibration between a depth camera and another device: the hand-eye transformation can either be estimated with a 2D image based approach or with the 3D measurements acquired by the depth camera.

2D and 3D data based hand-eye calibration For 2D color cameras, estimating the hand-eye calibration between the 2D camera and another device, such as robot or a coordinate measuring machine, is a well researched task [TL88] [SH06]. As most depth cameras also output a 2D intensity image in addition to the depth measurements, an obvious solution is to use the same algorithms for depth cameras as for 2D color cameras. For example, Reinbacher employed such an image based approach



(a) Kinect (b) SwissRanger 4000

Figure 4.3.: Depth cameras coupled with an industrial measurement arm [KK12].

for the hand-eye calibration between a depth camera and a robot [RRB12]. Kahn described an image based hand-eye calibration between a depth camera and an articulated measurement arm [KK12].

As depth cameras acquire dense 3D measurements, the hand-eye calibration can also be estimated based on these 3D measurements. For instance, Kim used the 3D measurement at the center of a marker for the hand-eye calibration [KH13]. Fuchs proposed a solution which uses depth measurements instead of 2D images [Fuc12]. This approach employs a calibration plane with known position and orientation. The hand-eye calibration is estimated by solving a least squares curve fitting problem of the measured depth values with the calibration plane. Furthermore, Kahn described a hand-eye calibration approach that aligns the depth cameras' 3D measurements with a 3D model of the calibration object [KHW14].

While both 2D and 3D data based approaches have been proposed, little is known about the accuracy and the suitability of these approaches for the hand-eye calibration with a depth camera. It is unknown whether 2D data based approaches have major advantages compared to 3D data based approaches (or vice versa), or whether both kinds of approaches can provide comparable results.

Therefore, Section 4.2.1 and Section 4.2.2 first describe both a 2D and a 3D data based hand-eye calibration approach [KK12] [KHW14]. These two hand-eye calibration approaches share a common main principle: first, the hand-eye transformation is estimated separately for each captured 2D or depth image. Then, the final hand-eye transformation is calculated by combining these separate estimations. Both approaches differ in the way the position and orientation of the depth camera is estimated: either by analyzing the captured 2D image, or by geometrically aligning the 3D measurements with a 3D

model of the calibration object. This algorithmic design choice makes it possible to directly compare the 2D image based and the 3D data based approach.

The main problem in view of the evaluation is, that the ground truth hand-eye transformation is not available and thus a direct evaluation of the accuracy is not possible. Therefore, Section 4.2.3 introduces quantitative 2D and 3D error measures that allow for an implicit evaluation of the accuracy of the calibration without explicitly knowing the real ground truth transformation. Based on these error metrics, Section 4.2.4 provides a comparative evaluation of both the 2D and the 3D data based hand-eye calibration.

4.2.1. 2D image based hand-eye calibration

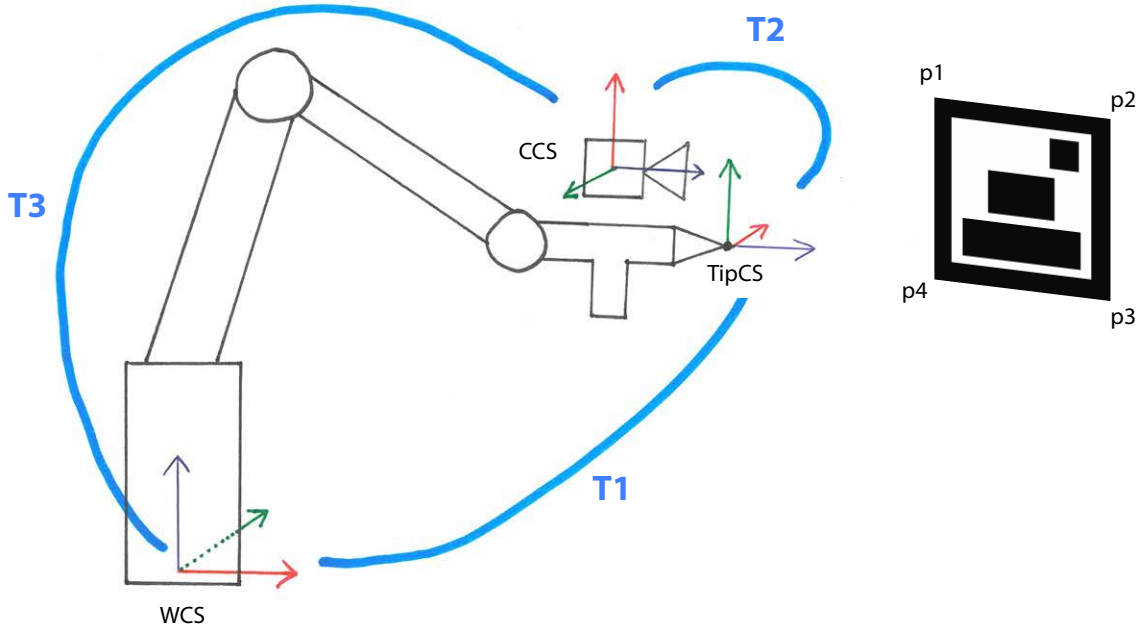


Figure 4.4.: Image based Hand-Eye Calibration.

Figure 4.4 visualizes a sketch of the measurement arm, the depth camera and an image marker which is used to calculate the hand-eye calibration with this image based approach. The world coordinate system (WCS) is defined such that it corresponds to the coordinate system of the measurement arm (the tracking coordinate system). The measurement arm outputs the transformation T1, which is the relative transformation between the measurement tip's coordinate system (TipCS) and the coordinate system of the base of the measurement arm (WCS). The transformation T2 is the hand-eye transformation between the coordinate system of the depth camera (CCS) and TipCS. T3 is the camera pose relative to the world coordinate system. Once the hand-eye transformation is known, the camera pose can be

calculated from the pose of the measurement arm and the hand-eye transformation with equation 4.1. In the notation of this equation, each transformation Ti is split up into its rotational and translational component (R_i and t_i).

$$\begin{aligned} R_3 &= R_2 \cdot R_1, \\ t_3 &= R_2 \cdot t_1 + t_2. \end{aligned} \tag{4.1}$$

The hand-eye transformation is calculated from n pose pairs ($T1_j$, $T3_j$) with $1 \leq j \leq n$. Each such pair contains a pose of the measurement arm's point tip and a depth camera pose, both relative to the world coordinate system. The main challenge for acquiring such a pose pair is the question how to calculate the pose of the depth camera $T3_j$ if the hand-eye transformation is not known yet. This task can be solved with the following two properties:

- The measurement arm can be used to measure the 3D coordinates of 3D points on object surfaces. The measured 3D coordinates are in the base coordinate system of the measurement arm (which is the world coordinate system).
- The pose of a camera can be calculated from a set of 2D-3D correspondences. Each such 2D-3D correspondence stores the position of a 3D point in the world coordinate system and its 2D projection onto the image coordinate system of the camera.

A 2D calibration pattern is used to obtain such 2D-3D correspondences. Here, the 2D calibration pattern is an image marker which can also be robustly detected with depth cameras which have a lower resolution than standard color cameras. This 2D calibration pattern is attached to a planar surface in the working range of the measurement arm and the 3D positions of its four corners ($p1$, ..., $p4$) are measured with the point tip of the measurement arm. The measured 3D coordinates are in the base coordinate system of the measurement arm (which is the world coordinate system). Then, the calibration pattern is detected in the 2D image captured by the depth camera.

The four 2D-3D correspondences (2D point in the image and the 3D coordinate of the detected 2D point in the WCS) as well as the intrinsic parameters of the depth camera and an image of the marker are the input for the camera pose estimation. Both the marker detection and the pose estimation are provided by the computer vision framework InstantVision [BBP*07]. The depth camera's pose $T3_j$ is estimated with direct linear transformation (DLT) and a subsequent nonlinear least squares optimization. Then, the accuracy of the calculated camera pose is estimated by a pixelwise comparison of the marker in the captured camera image with a simulated projection of the virtual marker image onto the 2D image, given the calculated camera pose. A pose pair ($T1_j$, $T3_j$) is only used for the hand-eye calibration if the projected marker matches the captured marker well in the 2D image. The equation used to calculate the hand-eye calibration $T2_j$ is specified in Equation (4.2) (it can easily be inferred from Equation (4.1)).

$$\begin{aligned} R_2 &= R_3 \cdot {R_1}^{-1} \\ t_2 &= t_3 - R_2 \cdot t_1 \end{aligned} \tag{4.2}$$

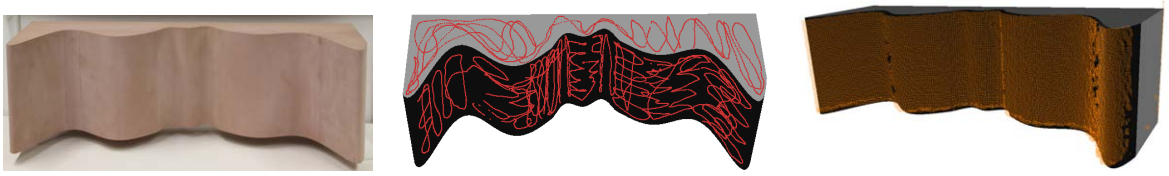
Theoretically, the hand-eye calibration could be approximated by a single pose pair. However, to improve the accuracy, many pose pairs are captured and $T2_j$ is calculated for each pose pair. Then, each rotational and translational parameter of the final hand-eye calibration is the median of this parameter in all collected $T2_j$ transformations. The median is used to calculate the final hand-eye transformation because it is more robust against outliers than the mean values.

4.2.2. Depth data based hand-eye calibration

The principle of the geometric hand-eye calibration is similar as the image based approach sketched in Figure 4.4. Just as for the image based approach, the transformation T1 is output by the measurement arm and T3 (the pose of the depth camera in the world coordinate system) is estimated for each single frame. Then, the hand-eye calibration T2 is estimated from T1 and T3 as specified by Equation (4.2). The difference between both approaches is that for the geometric approach, the pose of the depth camera (T3) is not calculated with image based camera tracking. Instead, it is estimated by geometrically aligning 3D measurements on the surface of the real calibration object (captured with a depth camera) with a virtual 3D model of the calibration object. Therefore, the geometric hand-eye calibration described in this section requires a 3D model of the calibration object.

Calibration object and 3D model Figure 4.5 shows a calibration object and a virtual 3D model of the calibration object. The calibration object was designed such that it accounts for the specific 3D measurement properties of depth cameras. The measurement accuracy of depth cameras depends strongly on the surface of the captured object. For instance, at jump edges or on object surfaces which absorb most of the light emitted by time-of-flight depth cameras, the measurement accuracy of these depth cameras is poor [Pia11] [SMAL13]. Therefore, the curved surface of the calibration object was designed such that no jumping edges occur on its front surface when the depth camera is moved in front of it. Furthermore, it consists of a material which diffusely reflects most of the light emitted by time-of-flight depth cameras and which thus supports the precision of the depth measurements. Additionally, the shape of the calibration object is designed in such a way that only one unique 3D alignment exists (neither symmetries nor periodicities).

Alignment of the virtual 3D model with the real calibration object Before the camera pose can be estimated with geometric alignment, as a preparation step, the virtual 3D model needs to be transformed such that it has the same position and orientation as the real 3D calibration object. To align the virtual 3D model with the 3D calibration object, sparse 3D measurements on the surface of the real 3D calibration object are acquired with the point tip of the measurement arm. Figure 4.5b shows such 3D points, colored in red. These 3D points are used for the alignment of the virtual 3D model with the real calibration object. The 3D point cloud and the 3D model are aligned with the Iterative Closest Point algorithm (ICP) [BM92] [RL01]. A point-to-triangle ICP variant is used, which iteratively reduces the distances between the 3D point cloud (measured on the surface of the real object) and the 3D triangle mesh of the virtual model. First, the 3D point cloud and the 3D model are coarsely aligned manually.



(a) Real 3D calibration object. (b) Virtual 3D model. (c) Virtual 3D model.

Figure 4.5.: 3D calibration object and 3D model of the calibration object, aligned with 3D measurements (red: acquired with the point tip of the measurement arm, orange: captured with the Kinect depth camera).

Then, the alignment is optimized with the ICP algorithm. In each iteration, the closest point on the triangle mesh is searched for each measured 3D point. Then, singular value decomposition is used to estimate a rotation and a translation which transforms the virtual 3D model, such that the average distance between both point sets is minimized. This iterative alignment reduces the average distance between the 3D points (consisting of 80.000 measurements) and the 3D model shown in Figure 4.5 to 0.2mm.

Camera pose estimation by geometric alignment The geometric alignment between a 3D point cloud and a 3D model is computationally expensive. Therefore, as a preparational step, an octree is created that hierarchically divides the space around the 3D model into rectangular regions. This speeds up the detection of closest points on the surface of the 3D model. Only those triangles need to be inspected which are located in the same region of the hierarchical bounding volume as the 3D point measured with the depth camera. For each captured depth image, the pose T3 of the depth camera is estimated with geometric alignment using the ICP algorithm.

The ICP algorithm requires a coarse initial estimation of the depth camera's pose. The transformation of T1 with the hand-eye transformation calculated with the image based approach provides such an initial estimation. An equally feasible approach would be to set the approximate camera pose for the first frame manually. Then, the hand-eye calibration calculated geometrically from previous frames can be used to initialize the camera poses of all other frames. Given the approximate pose of the depth camera, the following steps are repeated iteratively to improve the camera pose estimation with geometric alignment:

1. Render the 3D model with the current estimate of the camera parameters and use the rendered image as a validity filter. Reject all 3D measurements captured at pixels to which the 3D model does not get projected. This removes 3D measurements which do not belong to the surface of the calibration object.
2. Use the depth camera's pose estimation (R,t) with the following equation to transform each 3D measurements acquired with the depth camera from the camera coordinate system (p_{ccs}) to the

world coordinate system (p_{wcs}):

$$p_{wcs} = R^{-1}(p_{ccs} - t) \tag{4.3}$$

3. For each 3D measurement: Find the closest point on the triangle mesh (the octree speeds up this calculation).
4. Trim the found point pairs to remove outliers: reject those 5% of the found point pairs, which have the largest distance between the measured and the found 3D point.
5. Calculate the transformation that minimizes the distance between both point sets with singular value decomposition.
6. Update the estimated camera pose by applying the calculated transformation on the previously estimated camera pose.

Figure 4.5c shows 3D measurements captured with a Kinect depth camera, geometrically aligned to the virtual 3D model of the calibration object.

4.2.3. Error metrics

This section introduces error metrics that can be used for acomparative evaluation of hand-eye calibrations The quantitative evaluation of the hand-eye calibrations is subject to two major challenges:

1. The searched ("correct") hand-eye transformation is not known and cannot be measured directly.
2. The "correct" hand-eye transformation might be different for 3D measurements than for the 2D images captured with a depth camera. For example, the manual of the SwissRanger 4000 depth camera explicitly states that the 3D measurement's coordinate system is not located at the optical center of the depth camera [Mes09].

As no ground truth data is available for the hand-eye calibration, the accuracy of the hand-eye calibration needs to be evaluated indirectly (without comparing the estimated hand-eye calibration to "correct" reference values of the calibration). Furthermore, for applications which use both the 3D measurements and the 2D images acquired by a depth camera, the accuracy of the hand-eye calibration can not be assessed either with a 2D or with a 3D data based error metric alone: a solution, which is consistent with the 2D data, is not necessarily accurate in the 3D space (and vice versa). For these reasons, both a 2D and a 3D data based metric are used to evaluate the accuracy of the depth camera based hand-eye calibrations. Visualizations of both error metrics are shown in Figure 4.6.

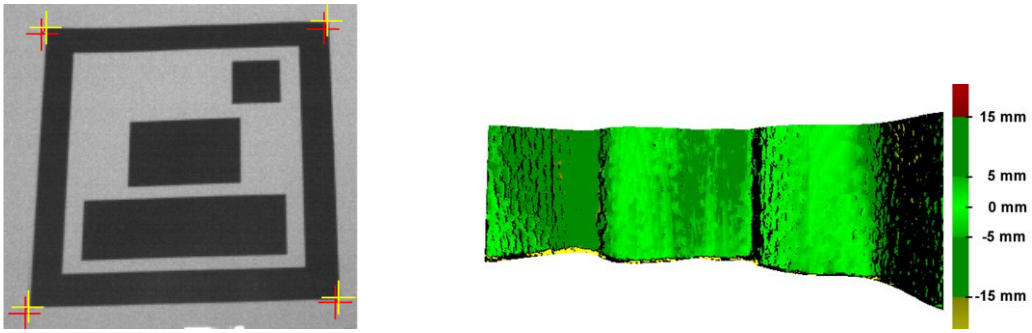
2D error metric The "normalized reprojection error" is used as 2D error metric. The unnormalized reprojection error measures the distance between the projection m of a 3D point M_{wcs} to the 2D image and the detected position of this point in the 2D image (m'). Here, M_{wcs} is the 3D position of a corner point of the 2D calibration pattern, measured with the point tip of the measurement arm as described in Section 4.2.1. For each frame of the evaluation sequence, the pose (R,t) of the depth camera is calculated from the pose of the measurement arm and the estimated hand-eye transformation with equation (4.1). Then, given the intrinsic camera calibration matrix K, the projection m of M_{wcs} onto the

2D image is calculated with

$$m = K[R|t]M_{\text{wcs}}. \tag{4.4}$$

The reprojection error increases when the camera is moved closer to the 2D calibration pattern. Thus, the projection error is normalized by the length of the 2D calibration pattern, to get the normalized reprojection error as a percentage of the calibration pattern's size. Given the projections m_i and m_{i+1} of two adjacent corner points of the calibration pattern, the normalized reprojection error (m_i, m_i') is:

$$\text{NReprojErr}(m_i, m_i') = 100 \cdot \frac{\|m_i - m_i'\|_2}{\|m_i - m_{i+1}\|_2}. \tag{4.5}$$



(a) 2D error metric (reprojection error in 2D image). Projected points m_i (red) and detected 2D points m_i' (yellow).

(b) 3D error metric: pixelwise difference between measured and real distance to the 3D calibration pattern.

Figure 4.6.: 2D and 3D error metrics.

3D error metric As 3D error metric, the distance between the 3D measurements of the depth camera and the surface of the calibration object is used. As described in Section 4.2.2, the 3D model used in this work was aligned with the real calibration project with an accuracy of 0.2mm. Thus, the 3D model provides ground truth data for the evaluation of the 3D measurements. To compare the depth measurements with this ground truth data, the camera pose is first calculated from the pose of the measurement arm and the estimated hand-eye calibration. Next, the 3D model is rendered from the current pose estimation of the depth camera. Then, the depth buffer values are compared with the depth values measured by the depth camera.

Please note, that even for a perfect hand-eye calibration, there are still 3D differences between the measured and the ground truth distance values. Such 3D differences are for example caused by measurement inaccuracies and systematic measurement errors of the depth camera. However, the total 3D error (caused both by inaccuracies in the hand-eye calibration and by other error sources) increases when the hand-eye calibration is inaccurate and decreases for accurate hand-eye calibrations. By using

the same evaluation sequence for both proposed hand-eye calibration approaches, the accuracy of both hand-eye calibrations can be directly compared.

4.2.4. Evaluation of hand-eye calibration: 2D or 3D?

The hand-eye calibrations was evaluated with a structured light depth camera (Kinect) and with a time-of-flight depth camera (SwissRanger 4000). The Kinect calculates distances by projecting an infrared pattern on the captured scene and by analyzing the distortions of the projected pattern. It outputs $640 \cdot 480$ depth values. In contrast, the SwissRanger emits infrared light and measures the time it takes for the emitted light to return to the camera after it has been reprojected by the captured scene. The SwissRanger 4000 provides $176 \cdot 144$ depth measurements.

Evaluation sequences The calibration and evaluation sequences were captured hand-held, by moving the measurement arm with the rigidly coupled depth camera around the calibration objects. The 3D sequences were recorded such that most of the front shape of the calibration pattern was captured: for frames in which only a small part of the 3D calibration surface is visible, an unambiguous alignment of the 3D measurements with the 3D shape of the calibration object can not be calculated. Furthermore, both for the 2D and the 3D calibration sequences, more images were captured such that the calibration object covered a rather large part of the image: both image based pose estimations as well as 3D depth measurements become less accurate with increased distances. The 2D calibration was detected in 3410 images of the Kinect infrared camera and in 5111 images captured with the SwissRanger 4000. For the geometric hand-eye calibration, 809 Kinect depth images and 2866 SwissRanger depth images were used.

Accuracy The results of the hand-eye calibrations are shown in Table 4.2 (Kinect) and in Table 4.3 (SwissRanger 4000). The SwissRanger captures less 3D measurements than the Kinect and the 2D image is more blurred and has a lower resolution. Therefore, the estimated camera poses vary more and the standard deviation is higher for the SwissRanger than for the Kinect depth camera.

Table 4.4 shows the accuracy as evaluated with the 2D evaluation metric (the reprojection error, see Section 4.2.3). Furthermore, Table 4.5 provides the results of the 3D evaluation metric. As noted in Section 4.2.3, the overall accuracy depends not only on the accuracy of the hand-eye calibration, but also on other factors such as the measurement accuracy of the depth camera. As the latter depends strongly on the distance between the camera and the captured object surfaces, the overall accuracy is specified for different ranges of measurement distances.

None of the two approaches (image based calibration and geometric calibration) is clearly more accurate than the other one. With the 2D evaluation metric, the image based calibration procedure performs better than the geometric hand-eye calibration (see Table 4.4). However, with the 3D evaluation metric, the geometric hand-eye calibration procedure performs better than the image based approach (Table 4.5). As explained in Section 4.2.3, the origin of a depth camera's 3D coordinate system is

not necessarily at the optical center of the camera. Therefore, in view of the accuracy of the hand-eye calibration for the 3D measurements, the 3D evaluation metric is more conclusive than the 2D evaluation metric. Thus, the 3D measurement based hand-eye calibration seems to provide a more accurate hand-eye calibration for the 3D measurements.

Distances in the calibration sequences For most measurement distances, the geometric hand-eye calibration provides more accurate results in view of the 3D measurements than the image based calibration (see Table 4.5). However, for very close distances, the accuracy is lower than with the calibration of the image based approach. This effect is probably caused by the distribution of the distances in the sequences used for the hand-eye calibrations. Figure 4.7 shows the calibration sequences' distance distributions of the camera centers to the 2D and the 3D calibration pattern. The accuracy is best for those distances with most input data. Due to the prerequisites in view of the visibility and the size of the calibration objects in the images, the 2D images were captured a bit closer to the calibration object than the data of the 3D calibration sequences. This effect is stronger for the Kinect data because the Kinect cannot measure depth values for surfaces too close to the camera. In order to acquire depth measurements of the whole 3D calibration object (without missing surface parts), most Kinect depth images were recorded with a distance of about 1m. Thus, for the Kinect, the 3D data based hand-eye calibration is most accurate for those distances at which the Kinect is best operated (at 1m distance, the Kinect does not suffer from missing surface measurements and acquires more precise depth measurements than for larger distances).

Kinect	**Image based calibration**	**Geometric calibration**
R	(-0.28, 0.80, 93.07)	(0.10, -0.27, -93.03)
std(R)	(0.82, 0.73, 0.22)	(0.71, 0.57, 0.47)
t	(13.30, -54.42, 80.48)	(22.07, -58.04, 93.23)
std(t)	(13.08, 10.10, 7.20)	(13.22, 5.76, 8.14)

Table 4.2.: Kinect: estimated hand-eye transformations (R,t) and standard deviations for Kinect depth camera. The rotation R is represented by a normalized axis angle, in degrees. The translation t is in mm.

Systematic depth measurement errors Depth cameras suffer from systematic depth measurement errors. This effect is shown by Figure 5.1 and is stronger for time-of-flight depth cameras than for the Kinect structured light depth camera. However, these systematic errors do not seem to have a strong effect on the accuracy of the hand-eye calibration, as the 3D data based hand-eye calibration also provides good results for the SwissRanger time-of-flight depth camera. This might be due to the symmetry of the systematic measurement errors, which might lessen systematic effects when aligning the 3D measurements with the 3D model of the calibration object.

SwissRanger 4000	Image based calibration	Geometric calibration
R	(1.36, 0.14, 89.87)	(0.17, 1.63, 90.10)
std(R)	(7.04, 6.80, 2.03)	(1.23, 1.29, 1.08)
t	(-11.63, 69.38, 103.68)	(-12.50, 40.80, 113.56)
std(t)	(18.27, 10.23, 13.00)	(15.93, 31.26, 6.02)

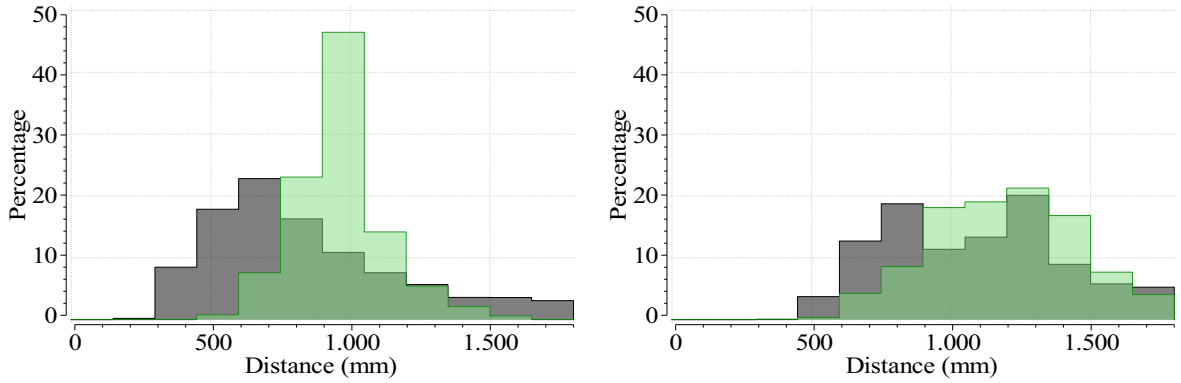
Table 4.3.: SwissRanger 4000: estimated hand-eye transformations (R,t) and standard deviations for SwissRanger depth camera. The rotation R is represented by a normalized axis angle, in degrees. The translation t is in mm.

Distance depth camera to surface	Kinect: image based calibration	Kinect: geometric calibration	SR4000: image based calibration	SR4000: geometric calibration
450-599	1.53	2.95	7.54	10.26
600-749	1.75	2.85	5.59	7.60
750-899	2.08	3.91	4.21	5.37
900-1049	2.34	5.13	3.44	4.08
1050-1199	2.75	6.55	3.29	4.62
1200-1349	2.86	7.77	3.67	5.67
1350-1499	2.96	9.14	4.79	7.21
1500-1649	3.20	10.56	6.21	8.87

Table 4.4.: **2D error metric:** Median of normalized reprojection errors. All values are in percent (ratio of reprojection error to the size of the 2D calibration pattern in the 2D image).

Distance depth camera to surface	Kinect: image based calibration	Kinect: geometric calibration	SR4000: image based calibration	SR4000: geometric calibration
450-599	3.70	13.01	8.90	19.05
600-749	4.88	12.35	10.17	16.81
750-899	6.87	4.84	11.42	12.58
900-1049	10.84	4.04	10.89	8.60
1050-1199	18.97	8.18	10.63	8.24
1200-1349	26.24	11.61	10.81	9.69
1350-1499	38.26	23.32	7.74	8.41
1500-1649	50.58	35.97	10.83	9.41

Table 4.5.: **3D error metric:** Median difference between the 3D measurements and the ground truth (3D position on the 3D model of the calibration object). All values are in mm.



(a) Kinect hand-eye calibration.

(b) SR4000 hand-eye calibration.

Figure 4.7.: Distribution of the distances from the camera centers to the calibration objects in the calibration sequences (histogram). Grey: image based calibration sequence, green: geometric calibration sequence. The histograms are drawn semi transparent (thus, at the dark green regions, both histograms overlap).

Combined 2D and 3D calibration To evaluate whether the accuracy of the hand-eye calibration could be improved by combining the image based and the 3D data based approach, three markers were attached on the wall above the 3D calibration object. The size of the markers was chosen such that they were fully visible when recording a sequence of the 3D calibration pattern. Then, for each frame, the hand-eye calibration was calculated both with the 2D images and with the 3D data. However, this combined approach neither increased the accuracy of the image based nor the accuracy of the 3D data based calibrations. The three markers covered only a rather small area of the image when both the markers and the 3D calibration pattern were visible in the same camera image, which decreased the accuracy of the image based camera pose estimations. Thus, the estimated camera poses were too inaccurate to improve the results.

Processing time The hand-eye calibrations were calculated with a 3.07 Ghz processor, using a single-core CPU implementation. For the Kinect, the estimation of the image based pose estimations used for the hand-eye calibration took 18 milliseconds per frame. The 3D data based camera pose estimations took 167 seconds per frame. For the SwissRanger 4000, the camera pose estimation times were 7 milliseconds per frame (image based), respectively 47 seconds per frame (3D data based). Thus, the 3D data based hand-eye calibration is much more computationally expensive than the 2D image based approach. On a CPU, the computation time is about one day for the 3D data based approach when 500 Kinect depth images are used. With the image based approach, the hand-eye calibration can be calculated in a few seconds.

4.3. Conclusion

In this section, first different approaches for pose estimation were discussed. Both image based pose estimation and geometric registration have major drawbacks for real-time 3D difference detection. With both approaches, the pose estimation fails or the accuracy decreases if not enough characteristic 2D features are visible in the captured 2D images, respectively if the captured 3D shape has one or two degrees of freedom in view of the depth alignment.

While the estimated camera position can differ by several millimeters to centimeters from the ground truth position if image based or depth image based pose estimation is used, a coordinate measuring machine provides an accuracy better than 0.1 mm. Furthermore, the accuracy of the pose estimation acquired with such a coordinate measurement machine does not decrease if the depth camera captures regions with few image features or 3D shapes which cannot be umambigously aligned. Thus, the proposed approach for 3D difference detection with high precision pose estimation is the combination of a depth camera with a coordinate measuring machine, such as a measurement arm.

For this reason, pose estimation with a coordinate measuring machine was detailed and comparatively evaluated in Section 4.2, with a focus on the estimation of the relative transformation between the depth camera and the coordinate measuring machine. Therefore, this section introduces and comparatively evaluates both a 2D image based and a 3D measurement based approach for the hand-eye calibration between a depth camera and a coordinate measuring machine. Both by this section and by the analysis and discussion of approaches for precise pose estimation in Section 4.1, this chapter provided an answer to question **Q2.1: "How can precise pose estimation be integrated in the 3D difference detection?"**

For depth cameras, the hand-eye transformation between the camera and a measurement arm can either be estimated using the 3D measurements or the 2D images captured by the depth camera. To compare both approaches, two hand-eye calibration algorithms were introduced which differ only in the way the camera pose is estimated (either 2D or 3D data based) and which are thus directly comparable. These algorithms were evaluated quantitatively, both with a 2D and with a 3D evaluation metric.

The quantitative evaluation shows that both the image based and the 3D data based hand-eye calibration algorithms provide accurate results. The 3D data based calibration provides more accurate results in view of the 3D measurements. However, this improved accuracy comes at the cost of the prerequisite that a 3D calibration object and an accurate 3D model of the calibration object are required. Furthermore, the surface of the 3D model needs to be sampled with the point tip of the measurement arm in order to align the 3D model and the calibration object. Thus, the 3D data based approach requires a more labour intensive preparation than the image based approach (for which it is sufficient to print a marker and to measure the four 3D coordinates of its corner points with the measurement arm).

Furthermore, the 3D data based hand-eye calibration is much more computationally expensive than the 2D image based approach. Thus, the 3D data based approach is well suited for applications which require very precise 3D data. In contrast, the image based approach provides a slightly less accurate, yet much faster and much less preparation intensive alternative.

5. Enhancing 3D difference detection by reducing measurement noise

As described in Section 2.2.3 and in Section 3.5, the measurement accuracy of depth cameras is reduced both by systematic measurement errors and by random measurement noise. Such measurement inaccuracies limit the accuracy of depth image based 3D difference detection. Therefore, this chapter addresses the question Q2.2:

Q2.2 How can measurement inaccuracies be reduced in the context of 3D difference detection?

First, different approaches for reducing the systematic and random measurement errors of 3D measurements acquired by depth cameras are discussed in Section 5.1, with a focus on the specific requirements of real-time 3D difference detection. Then, the integration of a real-time 3D reconstruction algorithm in the 3D difference detection is described in Section 5.2. This reconstruction algorithm reduces the measurement inaccuracies and removes gaps in the depth images at regions where no depth measurements could be acquired.

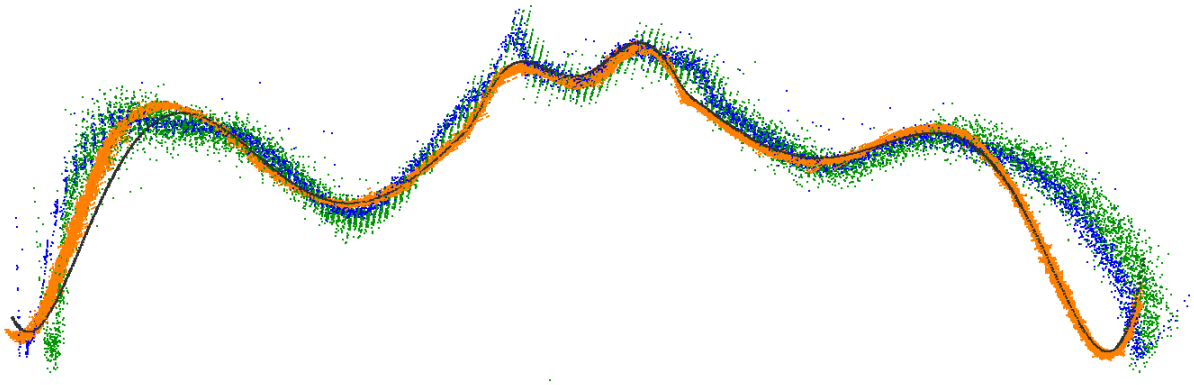


Figure 5.1.: Random measurement noise and systematic measurement errors, which cause a systematic difference between the shape of the real object and the measured shapes. Black: shape of the calibration object. Orange, blue and green: aligned 3D point clouds measured by different depth cameras (orange: Kinect, blue: SwissRanger4000, green: CamCube 3.0).

5.1. Discussion of approaches

Figure 5.1 visualizes both the random measurement noise and systematic measurement errors of 3D measurements acquired with depth cameras. The 3D points are the measurements of a single depth image. In order to provide a comparison of the shape of the 3D measurements with the shape of the captured surface, the 3D point clouds were aligned with the shape of the 3D model.

This section first introduces 3D calibration approaches for depth cameras in Section 5.1.1, which target the reduction of systematic measurement errors. In contrast, superresolution algorithms reduce the random measurement noise (Section 5.1.2). 3D reconstruction as described in Section 5.1.3 can be used to reduce both random measurement noise and to fill gaps of depth images.

5.1.1. 3D calibration of depth cameras

Depth calibration algorithms are used to estimate the distance measurement errors of depth cameras. Several approaches have been proposed for reducing the systematic measurement errors of specific depth cameras, both for different time-of-flight depth cameras [SBK08] [FH08] [KI08] [LSKK10] [BBGB*12] and for the Kinect depth camera [KE12] [CALT12] [SJP13]. These approaches have in common that the depth calibration is first estimated in an offline step. Then, during runtime, the measured depth values are adjusted according to the estimated depth calibration parameters. In order to estimate the depth calibration parameters in the offline calibration, the depth values measured by a depth camera are either compared to ground truth depth values or to reference depth values estimated with optimization algorithms.

Acquisition of reference values for depth calibration Lindner described the acquisition of reference depth values for depth calibration, both for a single depth camera and a setup which combines a depth camera with higher resolution color cameras [LSKK10]. For estimating reference depth values, a checkerboard calibration pattern is detected, either in the 2D image of the depth camera or in the 2D image of the color cameras. Then, the intrinsic and extrinsic parameters are estimated simultaneously with the parameters which model the depth measurement errors (parameters modeling wiggling related measurement errors and reflectivity related errors). Similarly, Fuchs et al. also estimate the extrinsic parameters and depth errors simultaneously [FH08]. Belhedi acquires reference depth values by estimating the pose of the depth camera relative to a flat wall [BBGB*12]. Therefore, the depth camera is coupled with a higher resolution color camera and the pose of the color camera relative to the wall is estimated with markers. Then, the pose of the depth camera is calculated from the pose of the color camera and a stereo calibration between both cameras.

Modeled depth calibration parameters Schiller et al. proposed a linear model for the correction of systematic depth measurement errors [SBK08]. They model the depth errors as a linear function which depends on the measurement distance and the pixel position of the depth measurement in the depth

image. The depth errors are modeled in relation to their pixel positions because the depth measurement accuracy of depth cameras often decreases with increasing distance from the image center. Belhedi et al. proposed a non-parametric model [BBGB*12], which also models the distoration variation according to the measurement distance and the pixel position of the depth measurement.

Kahlmann et al. estimate the depth measurement errors of time-of-flight cameras for different integration times [KRI06] [KI08]. They store the depth calibration results in a look-up-table, which is used to correct the depth measurements at runtime. In contrast, Lindner et al. [LK06] and Fuchs et al. [FH08] model depth measurement errors with splines. Fuchs introduced a calibration model for distance-related errors and amplitude related errors of time-of-flight depth cameras: every spline models the depth correction for a specific amplitude and a specific distance. Lindner estimates the depth calibration in two steps. First, a common depth distortion is estimated for the entire depth image. This distortion is modeled by a B-spline. Then, an additional per-pixel depth distortion is estimated for each pixel. In a more recent publication, Lindner et al. proposed a combined calibration approach that integrates the estimation of the extrinsic parameters, the intrinsic parameters and the depth calibration parameters in one combined calibration model [LSKK10]. They estimate parameters for adjusting the wiggling error and reflectivity related deviations.

Variable integration time For time-of-flight depth cameras, the depth measurement errors depend on the amplitude of the reflected light. However, the amplitude changes when the integration time of the light emitted by the time-of-flight camera is changed. Thus, as pointed out by Foix et al., it is difficult to apply depth calibration if the depth camera automatically adapts the integration time [FAT11]. In order to avoid overexposure or inaccurate depth measurements due to too few reflected light, an automatic adjustment of the integration time improves the measurement accuracy if the depth camera is moved. For a moving depth camera, the distances to the captured object surfaces change with each captured depth image, so a constant integration time either causes under- or oversaturation. Thus, it is not possible to use both an automatic adaption of the integration time and a depth calibration modeling the effects of the amplitude.

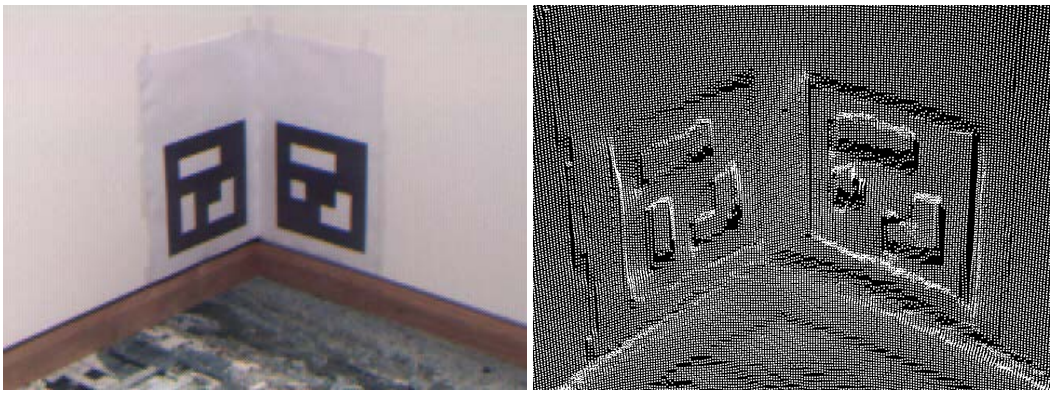
5.1.2. Superresolution

For static camera positions, the measurement noise can be reduced easily by capturing several depth images from the same static camera position and by averaging all depth values acquired at a pixel over time [SBSS08]. However, this is not possible for a moving depth camera because depth values acquired at a certain pixel correspond to different 3D points of the captured scene.

In contrast to simple averaging at pixels of depth images acquired with a static camera position, superresolution methods reduce measurement noise by combining several images taken from very close, but slightly different camera positions. While early superresolution methods were targeted at 2D images [FREM04], recently also superresolution methods have been developed for depth images. Depth image based superresolution approaches can be divided into two categories: first, methods which com-

bine a depth image with a higher resolution 2D image. Second, methods which fuse the depth data of several depth images.

Combination of depth data with a higher resolution 2D color image Superresolution approaches which use a higher resolution 2D image for enhancing the depth values of a depth image [GMZC09] [LHL11] [LMPD11] are based on the assumption that the depth values correlate with the color values of the 2D images. This assumption is met if depth edges in the depth image correspond to color edges in the 2D image and if uniformly colored regions of the 2D image correspond to depth image regions with similar depth values. To evaluate the applicability of superresolution methods which combine a depth and a color image, we implemented a depth-color based superresolution method [DT05] [Bri10]. For 3D objects for which the depth-color correlation assumption is met, the accuracy of the depth image could indeed be increased by the information from the 2D color image. However, in real scenarios, this assumption often is not met (for example if a 3D object is uniformly colored, or if a planar surface has different colors). In the latter case, the algorithm tends to estimate a relief in the depth image according to the color edges in the 2D image, thus decreasing instead of increasing the accuracy of the depth image. This effect is shown in Figure 5.2. Here, the edges of the color image imprint onto the depth image and incorrectly shift the 3D measurements of the planar surface.



(a) Color image. (b) 3D point cloud (detail of depth image).

Figure 5.2.: Superresolution artefact: the edges of the color image imprint onto the depth image.

Fusion of several depth images The second category of superresolution methods (which seek to enhance the depth accuracy by fusing several depth images) is not based on the color-depth correlation assumption and is thus better suited for enhancing the depth measurement accuracy in arbitrary scenarios [EOHM09] [KAM06] [KZ08] [STDT09]. While Paolini et al. [POPS09] and Nagesh et al. [NGL10] showed that it is in principle possible to execute superresolution algorithms on a GPU, currently all

superresolution algorithms proposed for fusing several depth images are offline algorithms, whose execution can take up to several minutes [CS11]. These computationally intensive offline algorithms are currently not suited for real-time applications with update rates of up to 30 frames per second. Furthermore, these algorithms pose specific requirements for the acquisition of the input depth images (in view of the feasible rotations and translations of the camera poses at which the input depth images are acquired). Thus, superresolution algorithms fusing several depth images can well be used to reduce the measurement noise of depth images in scenarios where the user can move the camera such that an optimal result is achieved by the superresolution algorithm [STDT09] [CSD*10]. However, superresolution algorithms are less suited for enhancing the accuracy of depth measurements in a real-time scenario where the user arbitrarily moves a hand-held depth camera around an object to inspect 3D differences from arbitrary viewpoints.

5.1.3. 3D reconstruction

In contrast to a static camera position, it is not possible to average 3D measurements over time for each pixel if the depth camera is moved. However, the depth information of several captured depth images can be merged by reconstructing a 3D surface with a 3D reconstruction algorithm. This can provide a more accurate surface estimation than the surface estimated from a single depth image. Similar to averaging over time for a static camera position, this merging step reduces measurement noise by fusing the data from several measurements.

While sparse 3D reconstruction is an algorithmic key component of real-time structure from motion or simultaneous localization and mapping [BBS07] [DRMS07] [KM09], dense 3D reconstruction is computationally very expensive. Thus, most approaches for dense 3D reconstruction reconstruct the 3D model offline, not in real time while the 3D data is acquired.

Rusinkiewicz provided one of the first solutions for real-time 3D model acquisition by separating the 3D model acquisition in two separate steps [RHHL02]. The first step is calculated in real time, while depth images are acquired by analyzing the projection of a stripe pattern onto the scanned object. This provides a live preview of the reconstructed 3D model. Then, the final 3D model is calculated in a second offline step. In the first step, the acquired depth image is aligned with the previous depth image with an efficient, projection-based variant of the Iterative Closest Points algorithm [BM92] [RL01]. Then, the aligned 3D points are discretized into the voxels of a voxel grid. Each voxel is rendered by a screen-aligned, semi transparent splat. While this approach provides a coarse preview of the scanned object, it does not merge the captured 3D data into a consistent surface representation. Thus, this real-time step does not provide a high quality. Therefore, an accurate 3D model is calculated in a second, offline step. In this offline step, the 3D point clouds are aligned with a slower ICP variant [Pul99] that provides a better accuracy than the ICP algorithm used for the real-time preview. Furthermore, the alignment is improved by a global alignment. Finally, a triangular mesh is extracted with the marching cubes algorithm [LC87].

Recently, several approaches have been proposed for dense real-time 3D reconstruction [PNF*08] [ND10] [SGC10] [NLD11] [NIH*11] [IKH*11]. In contrast to the approach proposed by Rusinkiewicz

[RHHL02], these algorithms do not separate the 3D reconstruction into a real-time and an offline step. Instead, they reconstruct the 3D surface at runtime. The algorithms proposed by Pollefeys [PNF*08], Newcombe [ND10] [NLD11] and Stuehmer [SGC10] reconstruct a 3D model from multiple 2D images. These algorithms estimate the camera pose and reconstruct sparse features with structure from motion, respectively with simultaneous localization and mapping. At the same time, they complement these sparse reconstructions with dense 3D reconstruction, using adaptations of dense stereo reconstruction algorithms. Newcombe et al. use an approximate 3D mesh reconstructed from the sparse structure from motion reconstruction to predict the view at reference frames. Then, this mesh is warped into depth maps acquired with view-predictive optical flow and constrained flow updates [ND10]. In order to estimate the camera poses, these approaches either use a sequence of 2D images [ND10] [NLD11] [SGC10] or combine 2D image based camera pose estimation with data from a Global Positioning System (GPS) and from an Inertial Navigation System [PNF*08]. Furthermore, an alignment of the captured 2D image with the reconstructed 3D model can improve the accuracy of the camera pose estimation [NLD11].

While these algorithms can reconstruct dense 3D data from 2D images, they suffer from the general drawbacks of image based 3D reconstruction: the reconstruction of a 3D surface requires the detection of characteristic 2D features in the captured 2D images. At untextured and featureless regions, the 3D shape of the reconstructed surface needs to be estimated with interpolation, which is based on the assumption that the 3D surface is smooth at the interpolated regions. Thus, the 3D reconstruction fails if this assumption is not met.

Recently, a depth image based 3D reconstruction algorithm has been proposed which fuses depth images acquired with a hand-held depth camera into a consistent representation of the captured object surface [NIH*11] [IKH*11] in real time. This algorithm is called KinectFusion. The real-time capability is achieved by a highly parallel execution of all calculation steps on the graphics card. Due to this massive parallelization, this 3D reconstruction algorithm can align and fuse depth images at a framerate of up to 30 frames per second. In contrast to superresolution methods, this approach does not reconstruct a single improved depth image but reconstructs an implicit 3D surface representation of the captured scene.

5.1.4. Discussion

The main advantages of the KinectFusion algorithm in comparison to superresolution algorithms are that the KinectFusion algorithm is real-time capable and that it does not restrict the way the user can move the depth camera to detect differences between the real object and the 3D model. In contrast, superresolution algorithms are not real-time capable and impose strict constraints of the input depth images: the input depth images need to be captured from very close camera poses. Thus, superresolution algorithms are not well suited for image sequences captured from arbitrary camera movement paths. Furthermore, in contrast to superresolution algorithms, the KinectFusion algorithm does not reconstruct a single improved depth image but reconstructs an implicit 3D surface representation of the

captured scene. Thus, the 3D reconstruction of the KinectFusion algorithm provides a smooth surface, not a single snapshoot from a single point of view.

For these reasons, an adaption of the KinectFusion algorithm was integrated in the 3D difference detection, in order to reduce the measurement noise of depth images and in order to fill the gaps in the depth images at regions where no depth data could be measured. In contrast to the KinectFusion algorithm, the captured depth measurements are not registered by geometric alignment but with the pose of the depth camera calculated with the external tracking device.

While a depth calibration is different for each camera, 3D reconstruction is independent of the specific camera. Furthermore, several depth cameras such as the SwissRanger 4000 depth camera already provide an automatic adjustment of the captured depth values with camera-specific calibration parameters. Therefore, the remainder of this chapter focusses on the reduction of random measurement noise with 3D surface reconstruction.

5.2. Enhancing the depth measurement accuracy with real-time 3D reconstruction

5.2.1. 3D reconstruction based on a truncated signed distance function

To reconstruct the 3D surface of a captured scene, 3D space is discretized into a discrete voxel grid. Each voxel of the grid stores the value of a "Truncated Signed Distance Function" (TSDF) at the center of the voxel. This value of the TSDF at the voxel center represents the distance of the voxel center to the closest reconstructed object surface. Points on the object surface have the distance 0. For 3D points with a non-zero distance, the sign specifies whether the voxel center is inside or outside the object surface. Reconstructing a 3D surface with a TSDF has the advantage that the 3D measurements of several captured depth images are merged for reconstructing the surface, which results in a more accurate surface estimation than the surface estimated from a single depth image. Similar to averaging over time for a static camera position, this merging step reduces measurement noise by fusing the data from several measurements of the same surface position.

Alignment with pose estimation by a tracking device In the KinectFusion algorithm, whenever a new depth image is acquired, the new depth image is geometrically aligned with the previously reconstructed 3D model. This alignment of the 3D data is based on a point-plane variant of the Iterative Closest Point algorithm [BM92]. However, as described in Section 4.1.2, geometric alignment has the drawback that the accuracy of the geometric registration depends on the shape of the captured object surfaces. If the registration of the 3D data is not umabiguous (for example due to a degree of freedom in the shape of the captured surface), the registration can drift. In this case, the new depth measurements are not correctly integrated in the 3D reconstruction. For example, this situation can occur if the captured depth images mainly contain planar surfaces of if the user moves the camera closer to the captured object, in order to inspect details. not only reduces A wrong or inaccurate alignment due

to ambuiguites in the captured shapes not only reduces the 3D reconstruction accuracy, but also the accuracy of the pose estimation of the succeeding depth images: these are again estimated by aligning the new depth images with the 3D reconstruction.

Thus, in order to avoid these problems, in this thesis, the pose estimation via geometric alignment is replaced by pose estimated with a tracking device. This replaces the geometric alignment step of the original KinectFusion algorithm. For example, if a coordinate measuring machine such as a measurement arm is used for the 3D difference detection, the depth camera pose estimation from the pose of the measurement arm is used to align the new depth measurements with the previously reconstructed 3D model.

After the alignment of the new depth image with the 3D model has been calculated, the data of the new depth image is merged with the estimation of the 3D surface reconstruction by updating the value of the TSDF for each voxel of the grid. The TSDF is an implicit surface representation which stores only the distances to the closest surface for discrete 3D points in space, but no explicit representation of the surface itself. However, the reconstructed 3D surface can be extracted from the implicit representation, either by the marching cubes algorithm [LC87] or by ray casting an artificial depth image from a specified (virtual) camera pose. While the marching cubes is not real-time capable, with ray casting an artificial depth image can be extracted in real time. For the real-time 3D difference detection approach described by this thesis, ray casting is used to extract a reconstructed depth image from the 3D reconstruction.

5.2.2. 3D difference detection with 3D surface reconstruction

The adapted 3D reconstruction algorithm was integrated into the 3D difference detection pipeline (see Figure 3.1), in order to to enhance the accuracy of the 3D difference detection by reducing the measurement noise of the captured depth images and in order to fill gaps of the depth measurements. To integrate this 3D reconstruction algorithm in the 3D difference detection pipeline, each depth image acquired by the depth camera is fed into the reconstruction algorithm. The new depth image is used to update the 3D surface reconstruction. Then, an artificial depth image of the current 3D surface reconstruction is created by ray casting. From the current pose of the depth camera, a ray is casted through each pixel of the virtual image of the depth camera. The depth value of this pixel is calculated by intersecting the view ray with the zero crossing of the TSDF. On a fast graphics card, both the update of the 3D surface reconstruction and the creation of the artificial depth image can be calculated in real time. Thus, instead of the real depth image captured by the depth camera, the artificial depth image from the 3D reconstruction algorithm is fed into the 3D difference detection algorithm. Then, the difference detection pipeline is executed the same way as it would for the original depth image acquired by the depth camera.

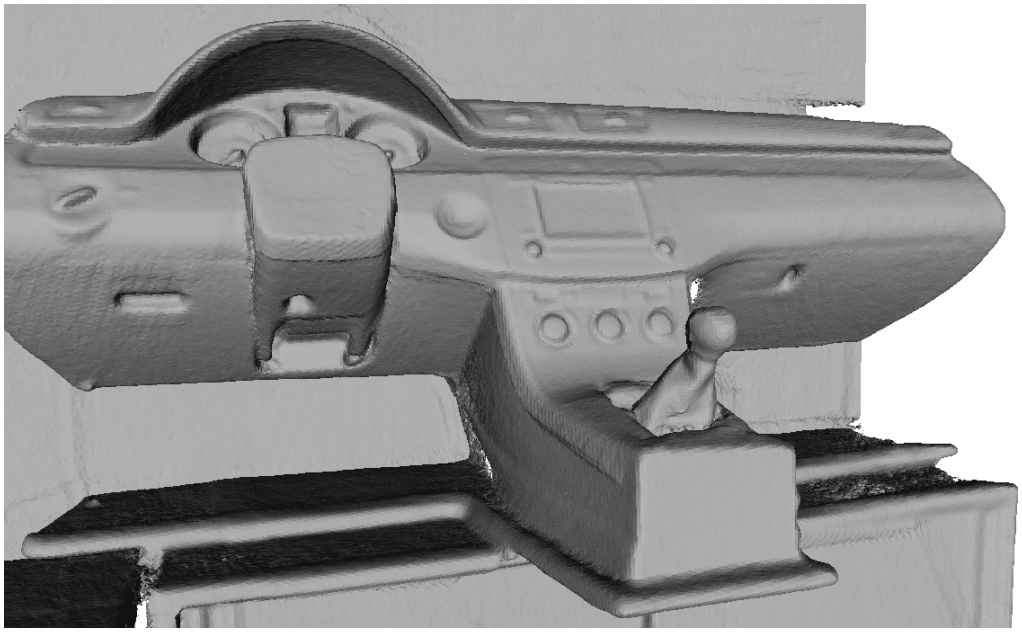


Figure 5.3.: Reconstructed control panel of a car.

5.2.3. Results

Figure 5.3, Figure 5.4 and Figure 5.5 show 3D objects reconstructed with the real-time 3D reconstruction algorithm integrated in the 3D difference detection. With this reconstruction, concave shapes can be reconstructed as well as convex shapes. The 3D reconstruction provides smooth surface reconstructions. However, it tends to smoth sharp edges as well. For example, in Figure 5.3, the buttons of the car cockpit are smoothed in the 3D reconstruction. Furthermore, in Figure 5.5, the hole in the planar rectangle on the right is a concave, curved surface in the 3D reconstruction. This smoothing is caused both by the depth images of the depth camera (the Kinect depth camera calculates a smooth distance estimation from the projected point pattern) and by the properties of the 3D reconstruction algorithm: the 3D reconstruction based on a truncated signed distance function estimates the distance to a surface for each voxel center, which smoothes the reconstructed 3D surface.

Figure 5.6 visualizes the accuracy of the 3D difference detection for the setup from Figure 4.2, both with and without the integration of the 3D surface reconstruction algorithm. The color scale on the right shows the color encoding of the measured differences (red: this pixel was measured closer than represented by the 3D model, yellow: this pixel was measured to be farther away than modeled). For example, the real gearshift differs from the modeled gear (both in view of its position and its shape) and the wheel is part of the 3D model, but not of the built mockup. A pixel is colored in black if the depth camera could not capture a depth measurement at this position (and thus, no 3D difference could be calculated). The pixels colored in dark blue are not part of the 3D model. The 3D surface

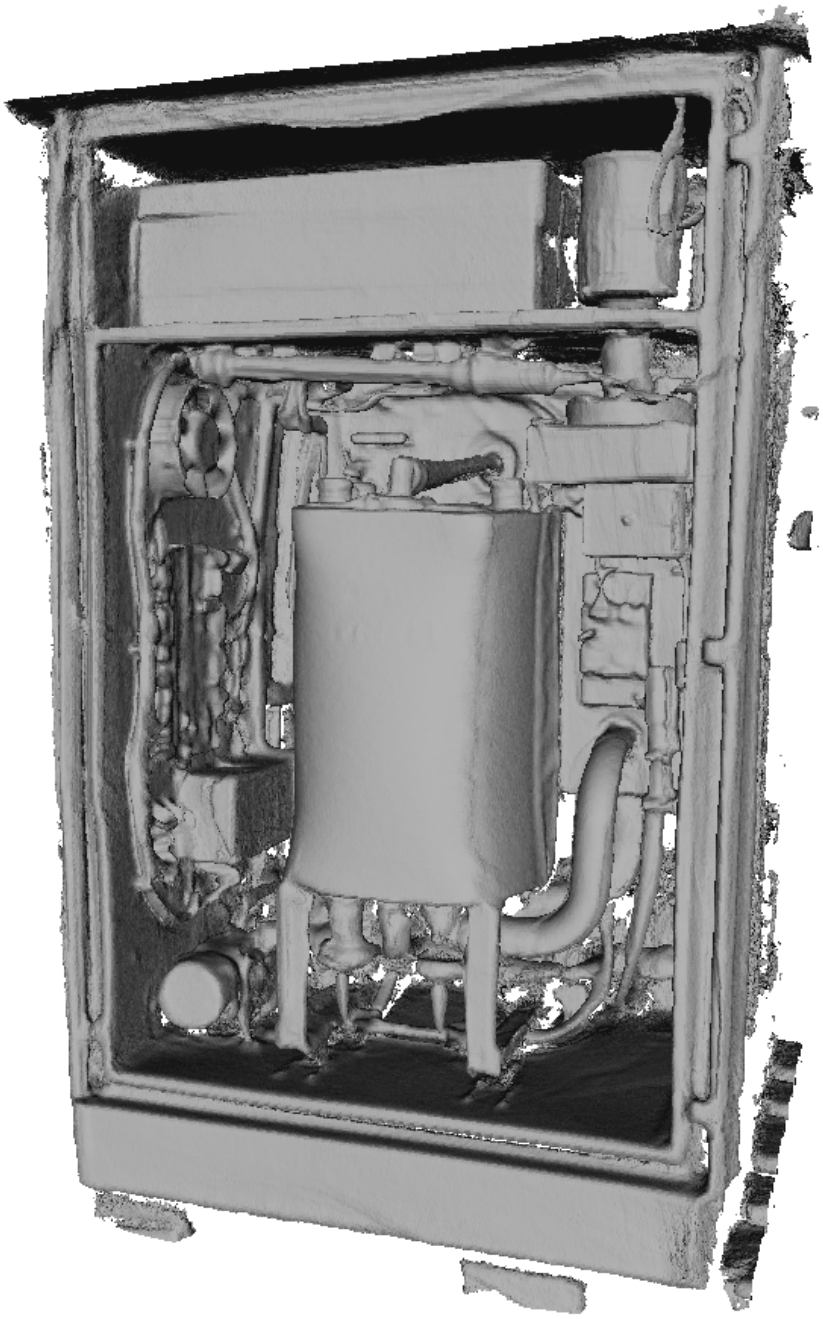


Figure 5.4.: Reconstructed fuel cell.

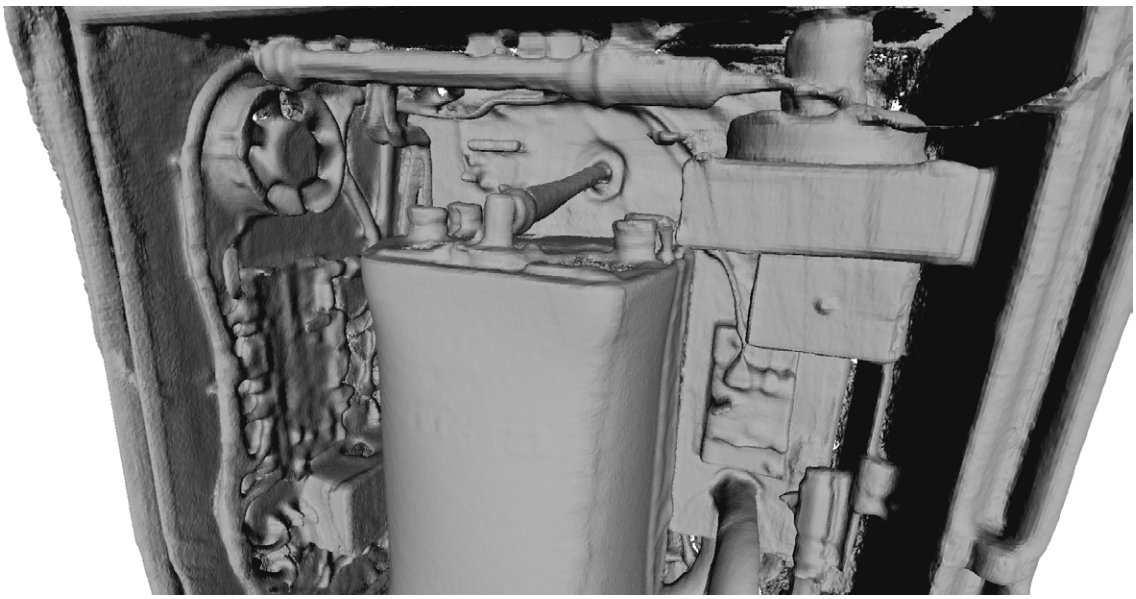


Figure 5.5.: Reconstructed fuel cell (detail).

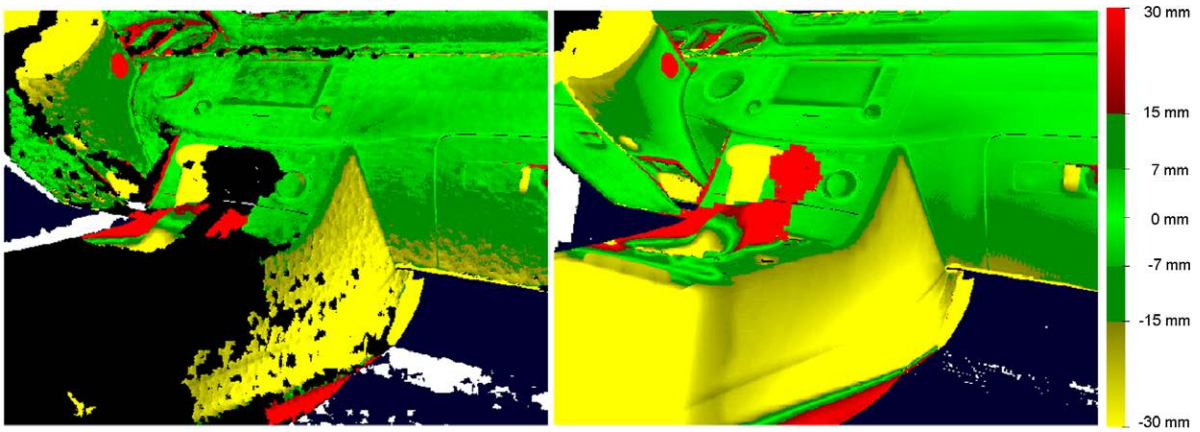


Figure 5.6.: 3D difference detection based on single depth image (left) and with reconstructed 3D model (right).

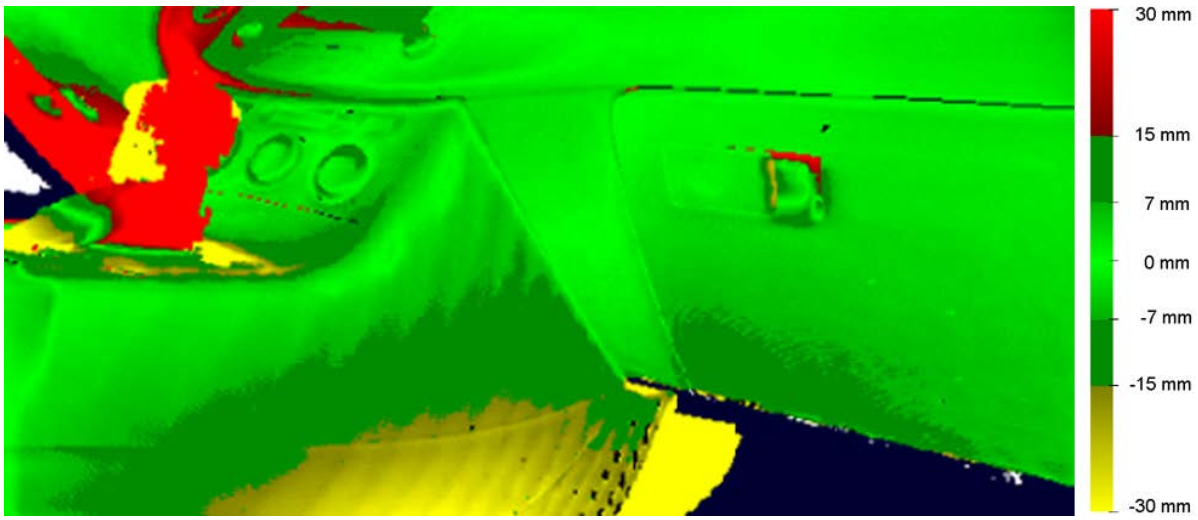


Figure 5.7.: The reconstruction of the side of the center part improves step-by-step when captured from a more orthogonal point of view.

reconstruction algorithm not only reduces the measurement inaccuracies on large parts of the measured surface but also reduces the areas for which no 3D difference could be calculated at all.

In Figure 5.6, the right side of the central element is colored in yellow although it does not differ much from the provided 3D model. This is because the previously recorded depth images were captured from a frontal view of the whole object. In the frame visualized in Figure 5.6, the camera has only just begun to move to the right. When the camera continues to be moved to the right (and to capture the right side of the central element from a more orthogonal point of view), the reconstruction of this part of the captured object improves frame-by-frame (see Figure 5.7).

Figure 5.8 provides a direct visual comparison of 3D difference detection without and with 3D reconstruction. The 3D reconstruction visually smoothes the 3D data and thus the 3D difference visualization. Furthermore, it provides a continuous 3D difference visualization of the 3D object surface, without the gaps in the difference visualization of the raw depth images at parts of the image where no depth data could be measured.

5.3. Conclusion

In Section 5.1, this chapter discussed different approaches for improving the accuracy of the captured depth images. While depth calibration approaches target a reduction of the systematic measurement errors of depth cameras, superresolution and 3D reconstruction reduce random measurement noise by fusing depth information from several depth images.

Superresolution algorithms provide a single depth image with improved depth accuracy, either by fusing depth information with a higher resolution 2D color image or by fusing depth data from several depth images captured from very close camera positions. In contrast to superresolution algorithms, 3D reconstruction (based on the discretization of the captured scene into a discrete voxel grid and the estimation of a truncated signed distance function) provides a smooth, continuous reconstruction of the surface. Furthermore, with a parallel implementation on a graphics card, the surface can be reconstructed in real time with 3D reconstruction, but not with state of the art superresolution algorithms. Additionally, superresolution imposes restrictions on the input camera poses (the depth images must be taken from varying, but very close camera positions) while the 3D images used as input for the 3D reconstruction can be taken from arbitrary camera poses. For these reasons, fusing several depth images by 3D reconstruction is better suited for depth image based 3D difference detection than superresolution algorithms. For these reasons, the 3D difference detection was extended by real-time 3D reconstruction.

In Section 5.2, the integration of real-time 3D reconstruction in the 3D difference detection was described. In contrast to 3D reconstruction based on geometric alignment, here the pose of the depth camera estimated with the tracking device (as described in Chapter 3 and in Chapter 4) provides the depth camera pose for the alignment of the depth measurements with the 3D reconstruction. For each new depth image, the measurements of the depth image are used to update the estimated 3D surface reconstruction, which is modeled by a discretized implicit function. Then, a reconstructed depth image is extracted via ray casting from the implicit representation of the reconstructed 3D surface. This reconstructed depth image replaces the depth image captured by the depth camera in the subsequent step of the 3D difference detection (comparison of the simulated depth image of the 3D model with the captured depth image). The qualitative results visualized in this section show that the 3D reconstruction provides smooth surfaces, both for convex and for concave surfaces. Furthermore, it fills gaps in the depth images at parts of the depth images where no depth measurements could be acquired.

Both by the discussion in Section 5.1 and by the extension of 3D difference detection with real-time 3D reconstruction, this chapter provided an answer to question **Q2.2: "How can measurement inaccuracies be reduced in the context of 3D difference detection?"**. A quantitative evaluation of the accuracy enhancement which is achieved by integrating the 3D reconstruction algorithm in the 3D difference detection is provided in the next chapter.

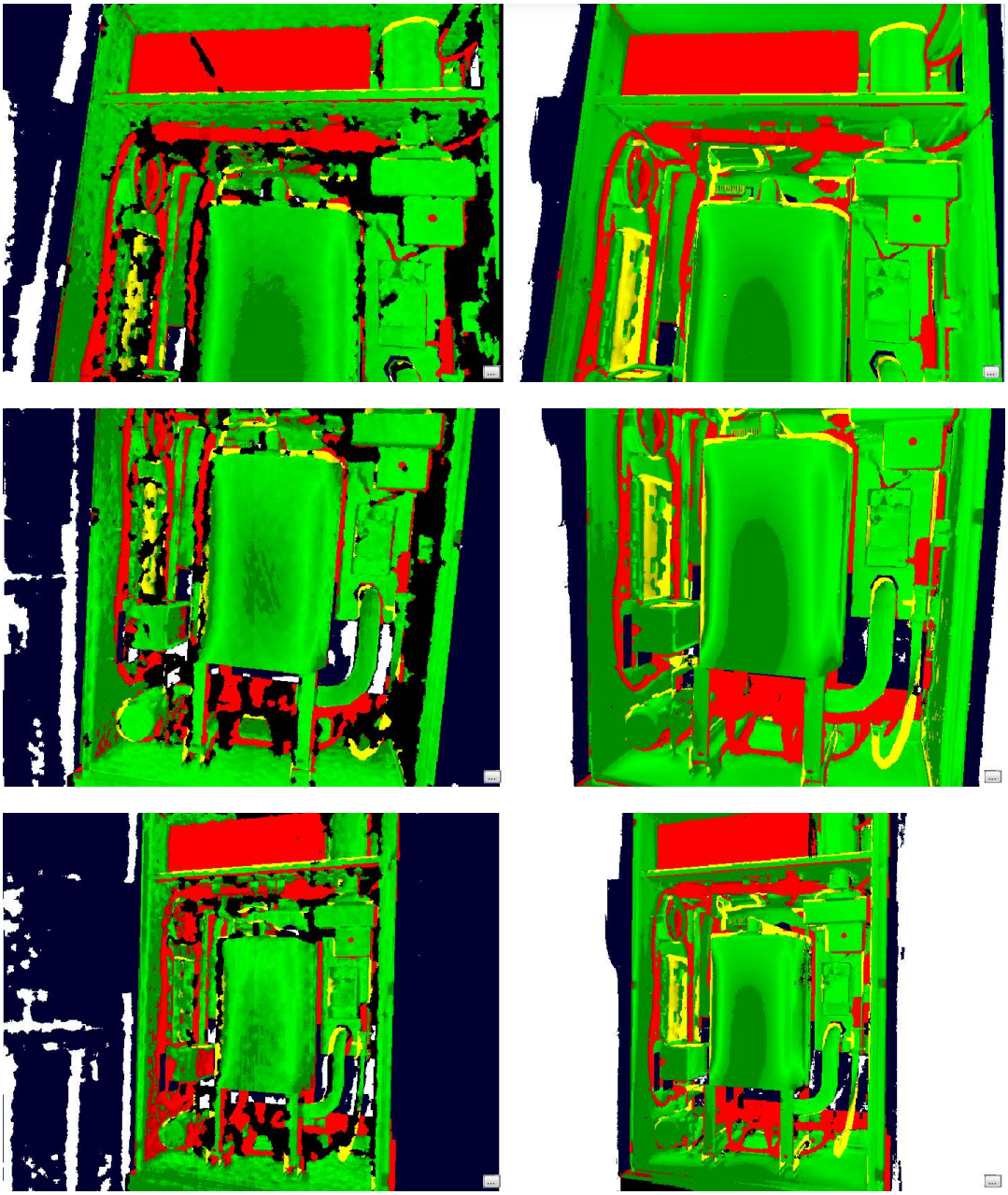


Figure 5.8.: Visualization of calulated 3D differences. Left: using a raw depth image (as acquired by the depth camera). Right: using the reconstructed 3D model for the difference detection.

6. Quantitative Evaluation

This chapter addresses the question:

Q3 ***Which accuracy can be achieved with concrete setups of the proposed concept for real-time, depth image based 3D difference detection?***

As pointed out in Section 1.3, this question is divided into two subquestions:

Q3.1 How do pose estimation inaccuracies and 3D measurement inaccuracies influence the overall accuracy of depth image based 3D difference detection?

Q3.2 Which accuracy can be achieved with different setups of the 3D difference detection, using real sequences captured with state of the art depth cameras?

First, Q3.1 is addressed in Section 6.1 by a simulation which quantifies the effects of different inaccuracies on the overall 3D difference detection accuracy. Then, Section 6.2 provides an evaluation of the proposed 3D difference detection for depth image sequences captured with state of the art depth cameras. The ground truth based, quantitative evaluation of these sequences provides answers to Q3.2 by quantifying the 3D difference detection accuracy, for the basic setup as well as for the variants proposed in Chapter 4 and in Chapter 5.

6.1. Simulation

In order to access the effects of different sources of inaccuracies on the 3D difference detection, their influence on the overall accuracy is quantified by a simulation [Kah13]. For the evaluation, the difference detection is applied on a depth image which is simulated with the correct parameters (reference image) in combination with a depth image which is generated with the modified extrinsic or intrinsic parameters (evaluation image). The following effects are evaluted by the simulation:

- Random measurement noise of the depth measurements.
- Inaccuracies in the estimation of the extrinsic depth camera parameters (rotation and translation).
- Inaccuracies of the estimated intrinsic camera parameters.
- Inaccuracies of the 3D model: approximation of curved shapes by planar surfaces.

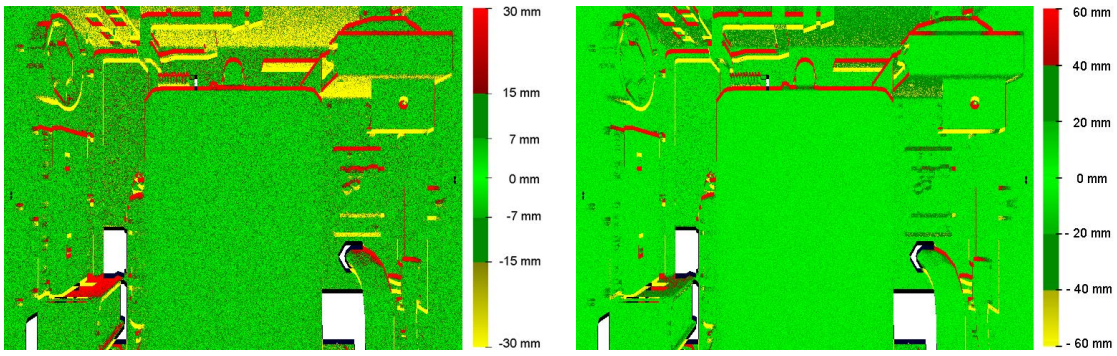
The influence of these parameters on the overall accuracy depends on the shape of the captured scene. For example, if the depth camera captures the shape of a plane which is orthogonal to the viewing direction of the depth camera, a movement of the camera parallel to this plane does not change

the measured distances. In contrast, if the surface is either not planar or not parallel to the viewing plane, the overall accuracy decreases if the estimated camera pose differs from the real camera pose.

For these reasons, the simulation is calculated for three different 3D shapes. First, the simulation is run for a planar surface orthogonal to the viewing direction of the depth camera. This simulation quantifies the effects of the accuracy of the estimated camera rotation and the accuracy of the camera position along the viewing axis. Second, a simulation is conducted with a 3D model of a fuel cell. This simulation exemplarily shows which effects different inaccuracies can have on the overall accuracy in a scenario with a complex industrial 3D object. Third, the effects of inaccuracies of the 3D model (approximation of curved shapes by planar surfaces) are evaluated with a simulation run from the origin of a unit sphere. For this setup, the simulation of a rotation quantifies differences caused by the approximation of the curved shape with planar triangles.

Depth measurement noise The depth measurements of depth cameras are affected by random noise as well as by systematic errors (see Section 2.2.3). The systematic errors are different for each depth measuring technology and are thus difficult to simulate. However, the effects of random measurement noise were evaluated. Therefore, gaussian noise with a standard deviation of 1% of the distance to the camera was added to the depth values. All simulations were run twice, once with and once without simulated measurement noise.

Figure 6.1 visualizes a simulation with simulated random measurement noise. The visualization is shown for two different color thresholds. Figure 6.1a shows differences with a stringent threshold and Figure 6.1b uses a less stringent threshold for the difference visualization. While the less stringent threshold shown in Figure 6.1b reduces the visualization of measurement noise, other differences are suppressed as well. For example, the differences on the upper part of the image (which are not caused by the random measurement noise) also become less distinguished. Thus, a reduction of the measurement noise supports the detection of smaller differences and visualizations with stricter thresholds.



(a) Bright green: <7mm, dark green: <15mm.

(b) Bright green: <20mm, dark green: <40mm.

Figure 6.1.: Simulation of 3D difference detection with random measurement noise.

6.1.1. Extrinsic parameters

The accuracy of the estimated depth camera position and orientation has a direct effect on the accuracy of the 3D difference detection. On the one hand, this accuracy is influenced by the accuracy of the pose estimated by the tracking device. Furthermore, if an external tracking device is used in addition to the depth camera, the accuracy also depends on the accuracy of the estimated relative transformation between the tracking device and the depth camera. Both error sources influence the accuracy of the estimated depth camera pose.

In order to quantify the influence of the accuracy of the depth camera pose, the simulated depth camera pose was varied while capturing images of a planar surface (XY-plane). The planar surface is orthogonal to the viewing direction of the depth camera and thus parallel to the depth camera's viewing plane. Thus, the x and y axes of the camera coordinate system are parallel to the plane and the z axis of the depth camera is orthogonal to the XY-plane. The distance of the camera to the XY-plane is one meter.

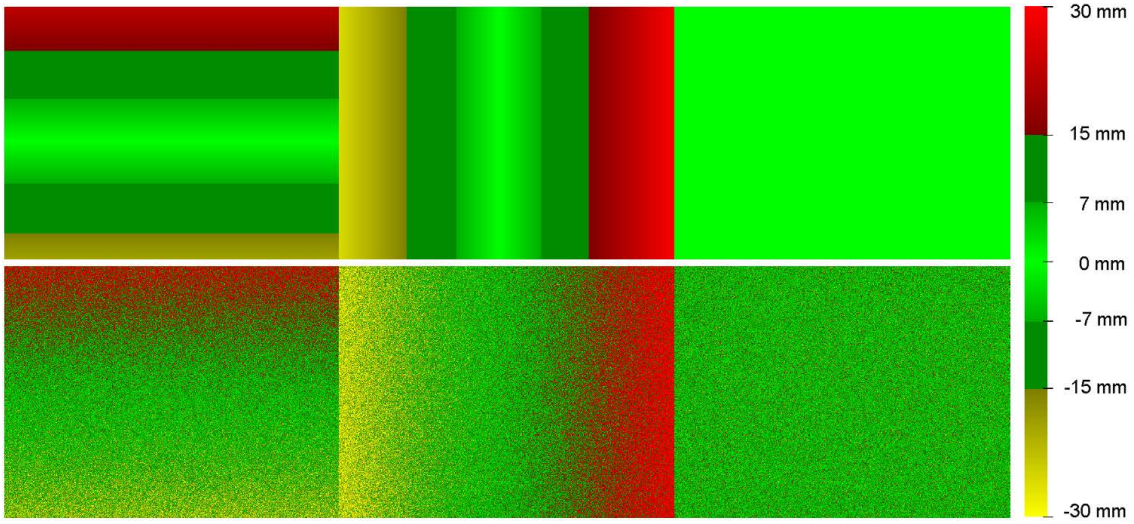
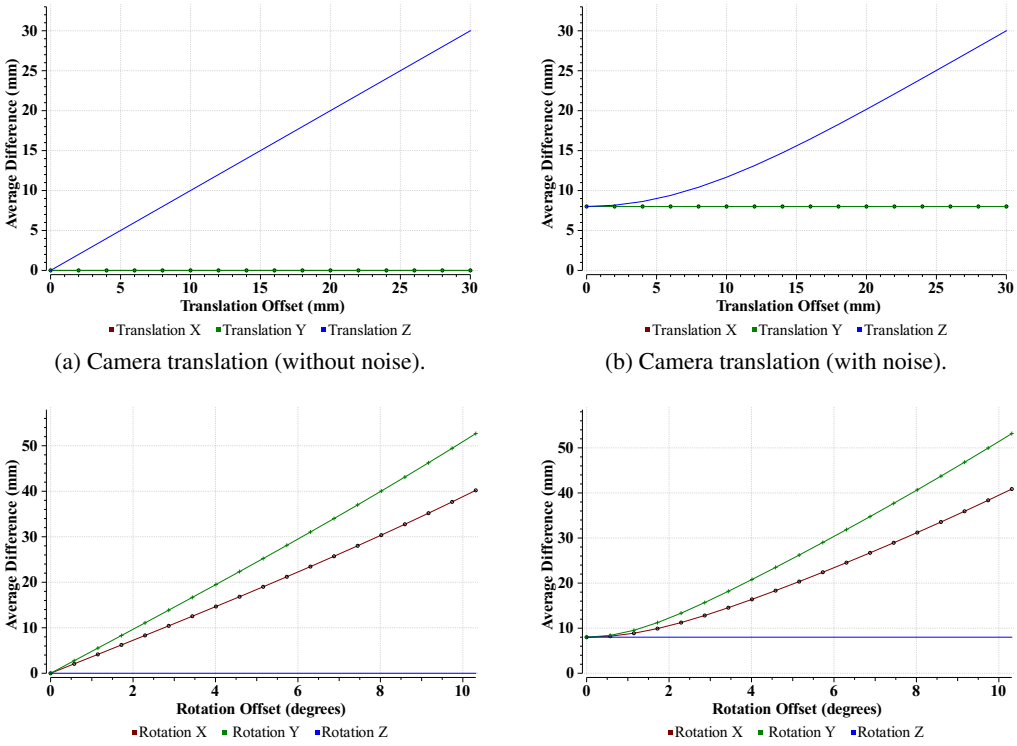


Figure 6.2.: Simulation of differences caused by inaccuracies in the camera pose estimation. From left to right: 3° offset in x rotation, 3° offset in y rotation, 3° offset in z rotation of the estimated depth camera pose. Top: without simulated depth measurement noise, bottom: with simulated noise.

Figure 6.2 visualizes the differences that arise if the estimated rotation of the depth camera differs from the real rotation. Furthermore, Figure 6.3 shows the quantitative results of this simulation. The y-axes of the plots represent the average difference between the "captured" (simulated) depth image and the depth image synthesized from the estimated camera pose. Thus, the plots show how this average difference increases subject to an increasing deviation between the real and the estimated extrinsic parameters. Each extrinsic parameter was evaluated independently. If the pose of the depth camera

is very accurately estimated, the accuracy of the calculation results is primarily influenced by the random measurement noise. However, if the pose error is significant (especially the rotational error), the difference detection accuracy is primarily reduced by the errors which are caused by the camera pose estimation. Please note that for the evaluation of the rotation offsets, the camera was rotated locally such that its position remained constant. If the extrinsic of the camera is denoted with (R,t), changing the rotation R while keeping the translation parameter t constant would not only change the rotation but also the camera position as well (because the rotation R is applied before t).



(a) Camera translation (without noise).

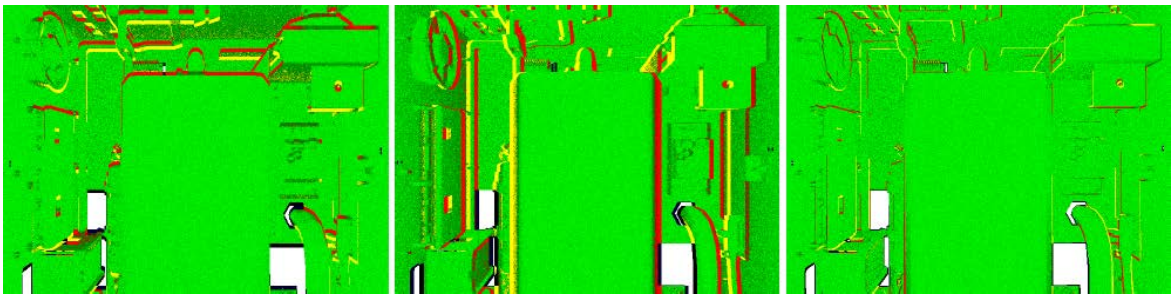
(b) Camera translation (with noise).

(c) Camera rotation (without noise).

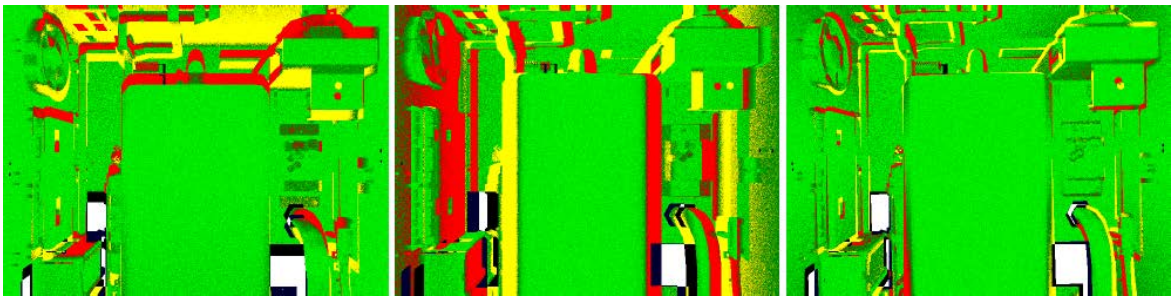
(d) Camera translation (with noise).

Figure 6.3.: **XY-plane:** influence of inaccurately estimated extrinsic camera parameters. For the XY-plane, inaccuracies of the intrinsic parameters (focal length and principal point) do not reduce the accuracy of the difference detection because the XY-plane is parallel to the viewing plane of the depth camera.

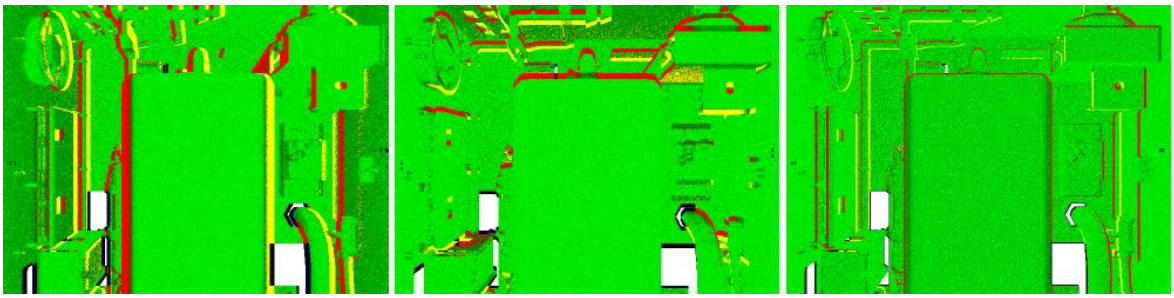
In addition to the quantitative evaluation with a simulated planar surface, Figure 6.4 exemplarily shows which differences can arise for an industrial object (a fuel cell), if the estimated extrinsic parameters differ from the real camera pose. Furthermore, Figure 6.5 provides the quantitative plots for this simulation.



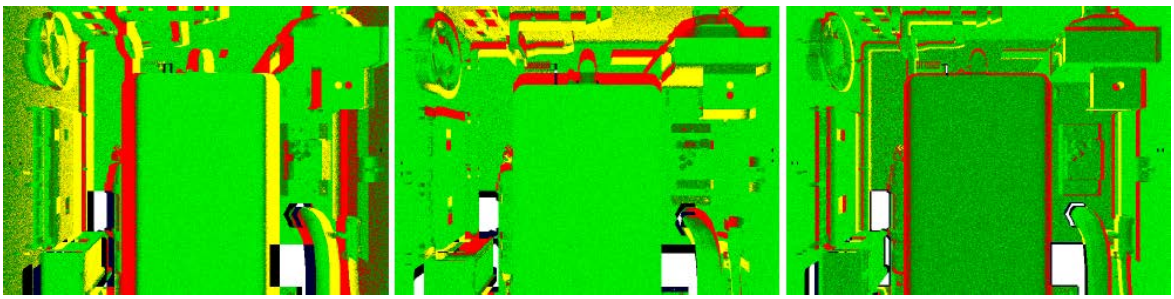
(a) 0.57° rotation offset.



(b) 1.7° rotation offset.

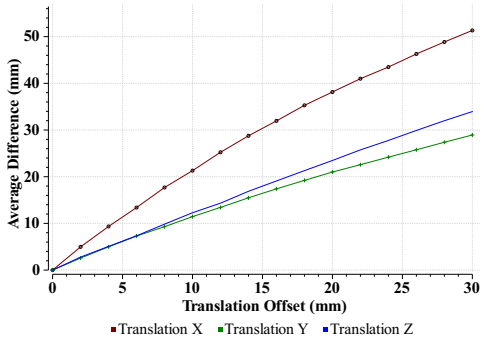


(c) 10 mm translation offset.

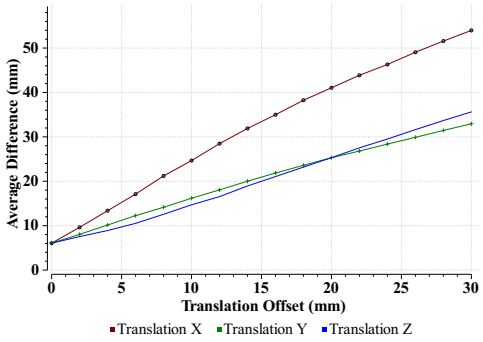


(d) 20 mm translation offset.

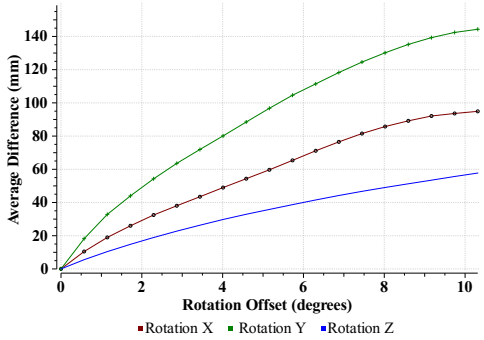
Figure 6.4.: **Fuel cell:** influence of extrinsic parameters (left: x-axis, center: y-axis, right: z-axis).



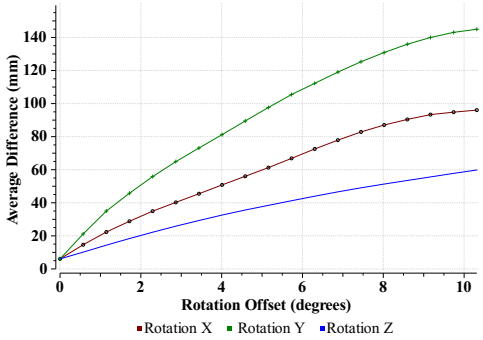
(a) Camera translation (without noise).



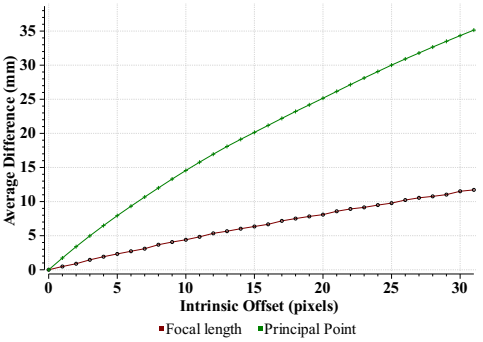
(b) Camera translation (with noise).



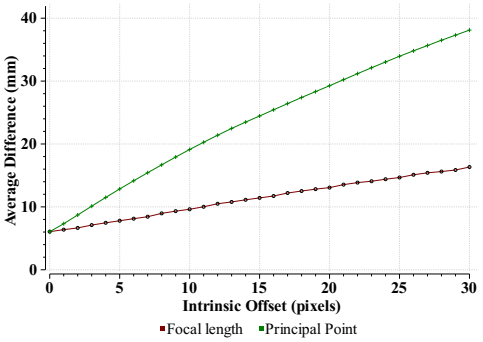
(c) Camera rotation (without noise).



(d) Camera translation (with noise).



(e) Incorrect intrinsic parameters (without noise).

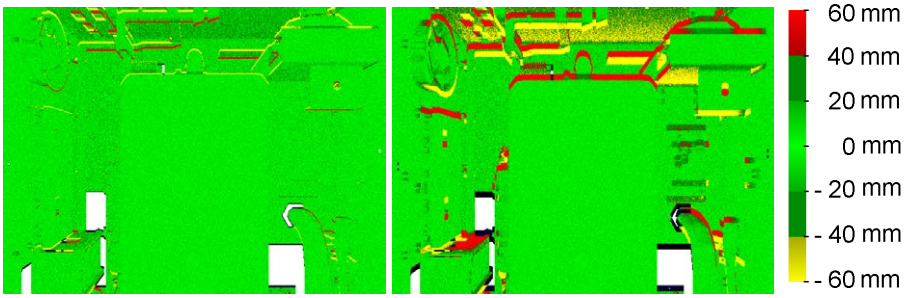


(f) Incorrect intrinsic parameters (with noise).

Figure 6.5.: **Fuel cell:** influence of inaccurately estimated extrinsic and intrinsic camera parameters. ((a)-(d)): Extrinsic parameters. (e)-(f)): Intrinsic parameters.

6.1.2. Intrinsic parameters

The third row of Figure 6.5 quantifies the influence of differences between the estimated intrinsic parameters and the actual intrinsic parameters of the depth camera. Furthermore, Figure 6.6 visualizes the effects of differences between the estimated and the actual focal length and principal point. An estimated principal point which differs from the actual principal point has nearly the same effect as an offset of the estimated camera position (see Figure 6.4d, center).



(a) Focal length offset (10 pixel) (b) Principal point offset (10 pixel)

Figure 6.6.: Influence of inaccurate intrinsic parameters (with simulated noise).

6.1.3. Inaccuracies of the 3D model

3D models often only approximate the modeled shapes. For example, polygonal meshes approximate curved objects with planar triangle surfaces. A standard VRML sphere with a radius of one meter was used to evaluate this effect. Figure 6.7 visualizes the polygonal mesh of the sphere, as well as the differences which arise if the sphere is rotated by 45° around its origin. For this visualization, the viewpoint was positioned at a distance of three meters to the origin of the sphere.

Figure 6.8 shows the differences that arise from the rotation of a virtual camera positioned at the center of the sphere. If the sphere model was smoothly curved, the rotational offset would always be 0. However, the polygonal approximation of the sphere causes differences between the depth images. Adding noise to the depth images reveals an interesting effect: The residual is larger than with either the noise or the approximation based differences alone, but not as much as the sum of both effects. In this case, the noise in the depth values smoothes the polygonal approximation of the curved surfaces, thus reducing the error caused by the triangle approximation.

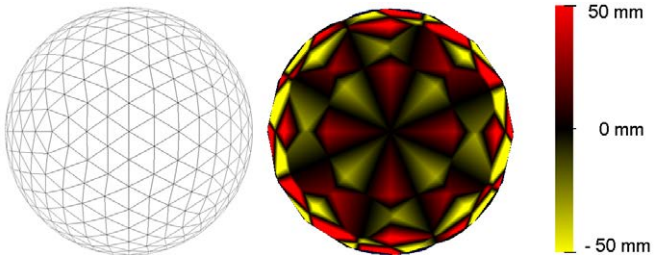
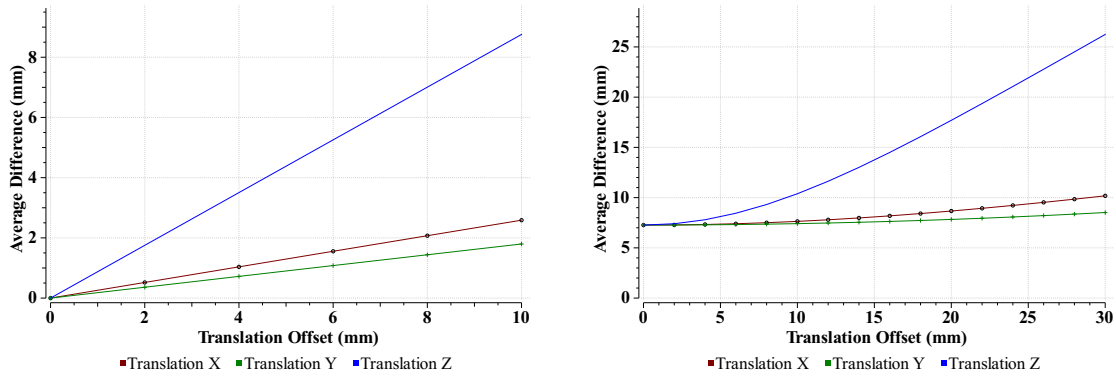
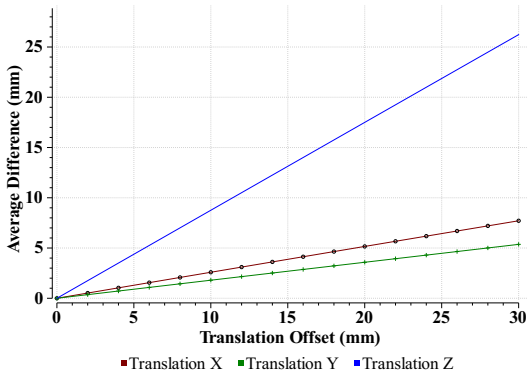


Figure 6.7.: Left: polygonal mesh of a 3D sphere. Right: difference visualization of two spheres with a radius of 1 m, rotated by 45°. The visualized differences arise from the approximations of the curved surface by the polygonal meshes, which are composed of planar triangles.

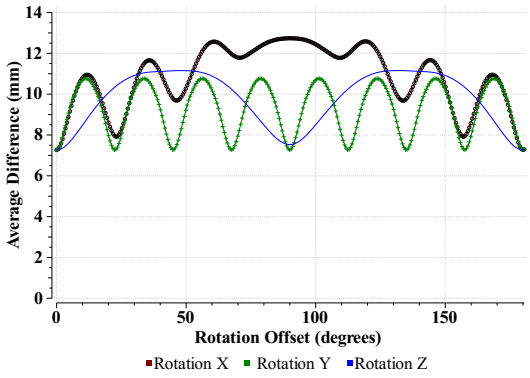


(a) Camera translation (without noise).

(b) Camera translation (with noise).



(c) Camera rotation (without noise).



(d) Camera rotation (with noise).

Figure 6.8.: **Sphere:** Influence of approximated 3D meshes on difference detection (the simulated camera position is at the center of a sphere with a radius of 1 meter).

6.2. Quantitative evaluation with input data acquired by depth cameras

While the simulation of the previous section provided a first theoretical estimation of the influence of the accuracy of the pose and the 3D measurements on the 3D difference detection, not all real-world error sources (such as the specific measurement characteristics of real depth cameras) can be simulated. In order to access the accuracy of depth image based 3D difference detection for real scenarios (such as industrial applications), a quantitative evaluation of the accuracy is required for depth image sequences captured with real depth cameras.

Therefore, this section provides a quantitative evaluation of the accuracy of depth image based 3D difference detection with different setups (image based camera pose estimation with an image marker, precise pose estimation with a measurement arm and depth measurements from a single depth image vs. fused depth measurements from a reconstructed object surface) [KBKF13].

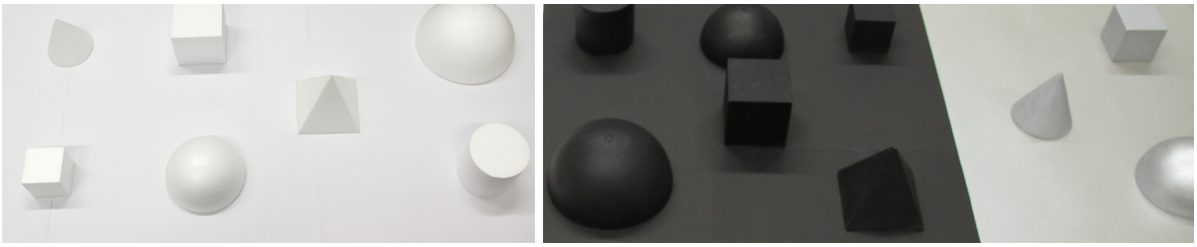
Depth Cameras The quantitative evaluation is based on test sequences recorded with state-of-the-art depth cameras (a SwissRanger 4000 time-of-flight depth camera and a Kinect structured-light depth camera). For the SwissRanger 4000 camera, drivers from MesaImaging [Mes13] were used and the data from the Kinect was acquired with the OpenNI interface [Ope13]. Both cameras were calibrated intrinsically. In addition to the intrinsic parameters of the pinhole camera model, the depth and color images were radially undistorted. The depth measurements were used as they were output by the depth cameras, they were not modified by a depth calibration.

Acquisition of Ground Truth Data The main challenge for conducting a quantitative evaluation of real measurements is the acquisition of ground truth data. For 3D difference detection, a prerequisite for the quantitative evaluation is the availability of a 3D model which exactly corresponds to the real object. For industrial objects, usually no 3D model is available which fulfills this requirement with the necessary precision. For example, the 3D model of the object shown in Figure 4.2 contains elements which are not part of the real object and vice versa and the 3D model cannot be remodeled easily. The 3D model of the pipes used for 3D difference detection in a previous publication [KWSF10] differs even more from the real pipes.

This is why the objects shown in Figure 6.9 were used for the quantitative evaluation. In comparison to other industrial test objects (such as the one from Figure 4.2), the 3D model of the industrial, metallic object from Figure 6.9a matches the real 3D object rather well. Furthermore, a 3D object with a curved surface was used (see Figure 6.9b) for which a very precise 3D model exists and a setup with several convex shapes (hemispheres, cubes, cylinders, cones and pyramids) with different surface colors (white, grey, black). These convex shapes were rigidly attached to planar boards. To create a 3D model of the combined evaluation object, first a separate 3D model was created for each convex shape. Then, the surfaces of the real objects were measured with the measurement tip of a Faro Platinum measurement arm. Finally, a point-triangle mesh variant of the Iterative Closest Point algorithm [BM92] was used to exactly align each 3D model with the measurement points on the real object. For the alignment, in

(a) Industrial, metallic evaluation object.

(b) Evaluation object with curved surface.



(c) Convex evaluation objects with different surface properties.

Figure 6.9.: Evaluation objects for quantitative ground truth evaluation.

total 115.000 3D points were measured on the surface of the convex objects with the measurement arm, 80.000 3D points on the object with the curved surface and 24.000 3D points on the outer surface of the metallic object. After the alignment, the average distance between the surface of the 3D model and the measurements on the surface of the real object was 0.3mm for the convex objects, 0.2mm for the object with the curved surface and 0.1mm for the metallic object.

For each evaluation object, depth image sequences were captured, both with a SwissRanger 4000 depth camera and with a Kinect depth camera. To acquire the evaluation sequences with a constant framerate, the framerate was limited to 10 fps (the SwissRanger depth camera automatically adjusts its integration time to the captured scene, so a higher framerate can only be achieved if the integration time is less than 100ms). Each Kinect depth image sequence consists of 500 to 800 frames, each SwissRanger 4000 depth image sequence of 4000 to 7000 frames. More images were captured with the SwissRanger depth camera to get an equivalent number of 3D measurements: while the Kinect outputs $640 \cdot 480$ depth values per frame, the SwissRanger has a resolution of $176 \cdot 144$ depth values. To compare the depth measurements with the ground truth data (the distance from the camera to the surface of the 3D model), the 3D model was rendered from the current pose of the depth camera. Then, the values of the graphic card's depth buffer were compared to the depth values measured by the depth camera. For each captured sequence, 10 to 38 million 3D measurements on the surface of the evaluation object were compared to their corresponding ground truth values from the rendered depth buffer.

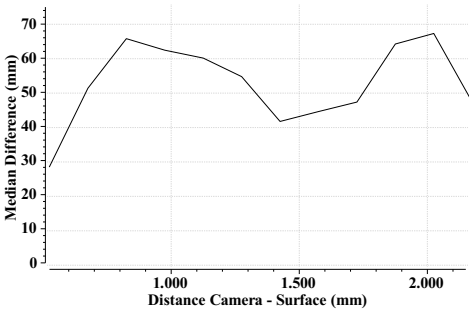
6.2.1. Pose estimation

The goal of this experiment was to quantify the accuracy of the 3D difference detection accuracy if the camera pose is estimated either image based (with a marker tracker) or with a measurement arm. For this experiment, the evaluation object with the curved surface of Figure 6.9b was used. In the marker tracking mode, the pose was estimated from the three square markers attached above the evaluation object.

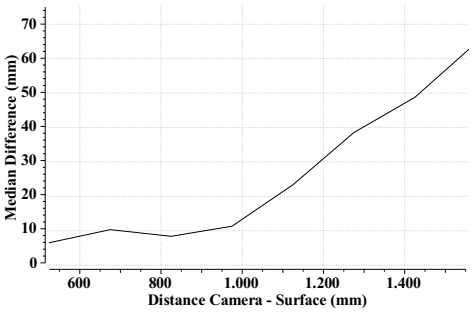
The Kinect contains both a color camera and a depth camera (see Section 2.2.2). For the marker-based pose estimation with the Kinect, the relative transformation between the color and the depth camera was calculated in an offline calibration. For evaluating the accuracy of 3D difference detection with a marker tracker for the Kinect, the markers were tracked with the $640 \cdot 480$ color camera of the Kinect. Then, the pose of the depth camera was calculated from the pose of the color camera by adding the previously calculated relative transformation to the pose of the color camera. In contrast to the Kinect, the SwissRanger depth camera does not contain an additional color camera but measures an intensity (grey) value for each captured depth measurement. The markers were tracked with this $176 \cdot 144$ intensity image. Figure 6.10a and Figure 6.10b show the overall accuracy of the 3D difference detection for image based camera pose estimation with a marker tracker.

To estimate the accuracy of 3D difference detection if the pose of the depth camera is measured with a measurement arm, in a second step the same sequences were evaluated with a measurement arm based pose estimation. The pose of the depth camera was calculated from the pose of the measurement arm by adding the previously calculated hand-eye calibration (between the tip of the measurement arm and the depth camera) to the measured pose of the measurement arm. Figure 6.10c and Figure 6.10d visualize the overall accuracy of the 3D difference detection for depth camera poses estimated from the pose measurements of a measurement arm (which is a coordinate measuring machine, CMM).

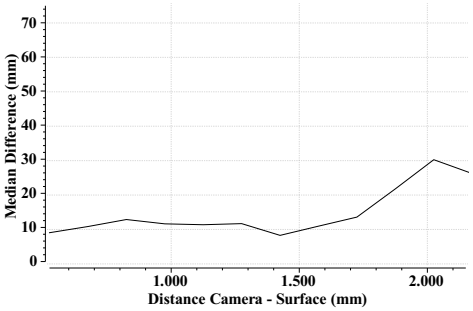
The measurement arm based pose estimation increases the accuracy for both depth cameras. This effect is more pronounced with the SwissRanger depth camera because the marker based pose estimation of the SwissRanger is less accurate than the marker based pose estimation of the Kinect camera. This is due to the lower resolution of the SwissRanger's 2D image used for the marker based camera pose estimation. The accuracy of the 3D difference detection with the Kinect depth camera is increased as well.



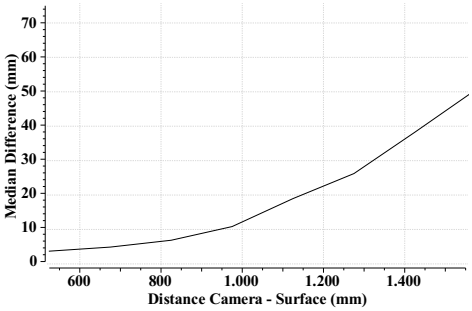
(a) SwissRanger 4000, marker based pose estimation.



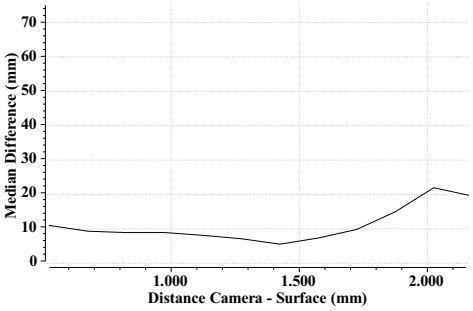
(b) Kinect, marker based pose estimation.



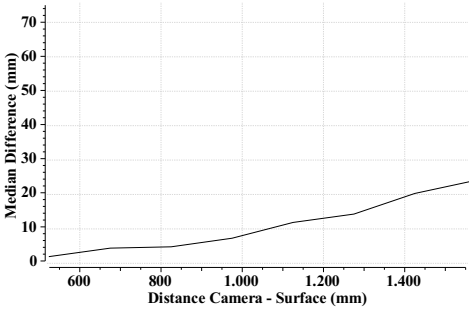
(c) SwissRanger 4000, CMM pose estimation.



(d) Kinect, CMM pose estimation.



(e) SwissRanger 4000, CMM pose estimation, diff. detection based on reconstructed 3D model.



(f) Kinect, CMM pose estimation, difference detection based on reconstructed 3D model.

Figure 6.10.: 3D difference detection accuracy for marker based pose estimation, measurement arm (CMM) based pose estimation and CMM based pose estimation in combination with the 3D surface reconstruction algorithm. Evaluation object: curved surface (Figure 6.9b).

6.2.2. 3D surface reconstruction

This experiment was conducted to evaluate whether the accuracy of 3D difference detection can further be improved if the 3D difference detection is based on a reconstructed 3D model of the captured object (instead of 3D measurements from a single captured depth image). For this purpose, a real-time 3D reconstruction algorithm was integrated in the 3D difference detection pipeline as described in Section 5.2. Figure 6.10e and Figure 6.10f show the overall accuracy of the 3D difference detection which is achieved if the camera pose is estimated with a measurement arm in combination with the 3D surface reconstruction algorithm. The combination of precise pose estimation with 3D reconstruction provides the best accuracy. With this setup, the accuracy is better than for 3D reconstruction without precise pose estimation by a measurement arm and better than for the setup which uses only precise pose estimation (without 3D reconstruction).

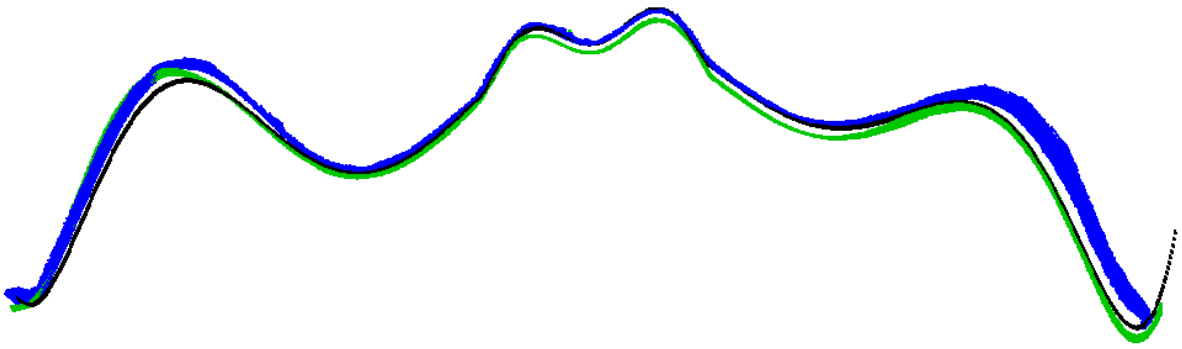


Figure 6.11.: Offset between 3D reconstruction and ground truth 3D surface of reconstructed object (Kinect). Black: 3D surface of virtual 3D model. Green and blue: Reconstructed 3D surfaces, from two different points of time of the 3D reconstruction (green: first 240 depth images, blue: 790 depth images).

While the 3D reconstruction reduces the random measurement noise, systematic 3D measurement errors (as visualized in Figure 5.1) are not leveled out. These systematic measurement errors cause distortions in the shape of the reconstructed 3D surface. This effect is illustrated in Figure 6.11. In contrast to Figure 5.1 (which shows single depth images aligned with the shape of the virtual 3D model), Figure 6.11 visualizes the shape of the reconstructed 3D surfaces. The 3D reconstructions are shown for two different points in time: the 3D reconstruction based on the first 240 depth images is shown as well as the 3D reconstruction based on all 790 depth images.

For both 3D reconstructions, the reconstructed shape partially differs from the actual shape of the captured object. The differences between the two 3D reconstructions are caused by the different camera positions of the acquired depth images. In the first part of the sequence, the depth camera mostly captured the object surface from viewpoints on the right. Later, the depth camera captured the object more from the left (here, right and left refer to the point of view of the depth camera capturing the

object surface). The different points of view of the depth images caused different systematic measurement errors, and thus different shapes of the 3D reconstructions. Please note that the evaluation of this chapter uses the raw depth measurements (as captured by the depth cameras) as input for the 3D reconstruction. The depth measurements are not adjusted by a depth calibration as described in Section 5.1.1. Thus, in future work, it might be possible to increase the accuracy of the 3D difference detection further by integrating a depth calibration in the 3D difference detection, in order to reduce the systematic measurement errors of the depth cameras. Table 6.1 shows that the 3D reconstruction improves the accuracy of 3D difference detection with a Kinect in spite of the influence of the systematic measurement errors on the reconstructed 3D shape.

6.2.3. Comparison of accuracies

Table 6.1 provides the numerical values of the 3D difference detection accuracy with the Kinect depth camera for the marker based pose estimation and for the pose estimation with a measurement arm, both with and without the accuracy enhancement by the 3D surface reconstruction. Both the measurement arm based pose estimation and the 3D reconstruction algorithm improve the overall accuracy of the 3D difference detection. The best accuracy is achieved when these two approaches are combined.

Distance Kinect camera - surface	Marker pose without 3D rec.	Marker pose with 3D rec.	CMM pose without 3D rec.	CMM pose with 3D rec.
450-599	6.54	7.76	3.70	1.96
600-749	10.34	10.71	4.88	4.41
750-899	8.40	6.50	6.87	4.80
900-1049	11.34	8.54	10.84	7.30
1050-1199	23.39	13.37	18.97	11.88
1200-1349	38.56	22.81	26.24	14.31
1350-1499	48.78	39.85	38.26	20.31
1500-1649	64.49	48.35	50.58	24.11

Table 6.1.: Median difference between 3D measurements and ground truth (numerical values of Figure 6.10 for difference detection with a Kinect depth camera). Additionally, the results of the setup "marker pose with 3D reconstruction" are provided, which can be used without a measurement arm (CMM). All values are in mm.

In the column "marker pose with 3D reconstruction", Table 6.1 provides the evaluation of a setup which does not require a measurement arm. In this setup, the camera pose is estimated with a marker tracker and the accuracy of the 3D difference detection is enhanced with the 3D surface reconstruction algorithm. Compared to pose estimation with a CMM, this setup has the drawback that the marker always needs to be visible in the image of the 2D camera. For marker based pose estimation with a Kinect, the accuracy of the 3D difference detection was improved by integrating the 3D reconstruction algorithm. In contrast to the Kinect, the 3D difference detection of the SwissRanger 4000 could not be

improved by the 3D reconstruction algorithm when a marker tracker was used for the pose estimation. As the pose can only be estimated very coarsely with a marker tracker on the low resolution intensity image of such a depth camera, most of the errors visualized in Figure 6.10a were caused by errors in the pose estimation. Thus, they can not be smoothed out by the 3D surface reconstruction algorithm, which requires pose estimation to reconstruct the 3D surface.

6.2.4. Comparison of the 3D measurements with the self-reconstructed 3D model

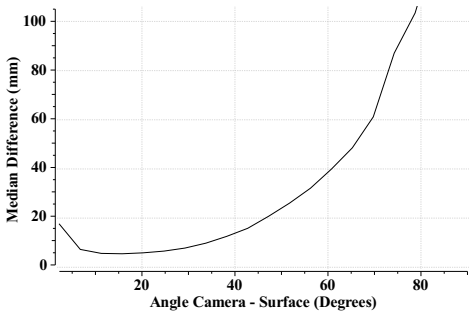
Table 6.2 visualizes the differences which arise if the ground truth 3D model (shown in Figure 4.5) is replaced by the 3D model which was reconstructed as described in Chapter 5.2. Here, the differences are much smaller than the differences between the 3D measurements and the ground truth 3D model (see the last two columns of Table 6.1). This is due to the effect that the systematic measurement errors of the Kinect depth camera affect the shape of the 3D reconstruction (see Section 6.2.2 and Figure 6.11). Thus, the reconstructed 3D model matches the 3D measurements of the depth camera better than the actual ground truth 3D model. The same is true with and without the reduction of measurement noise by on-the-fly 3D reconstruction during the 3D difference detection.

Distance Kinect camera - surface	**CMM pose without 3D rec.**	**CMM pose with 3D rec.**
450-599	3.14	1.81
600-749	4.18	3.47
750-899	3.92	4.00
900-1049	6.13	4.08
1050-1199	11.79	5.21
1200-1349	20.87	7.93
1350-1499	29.31	9.77
1500-1649	38.67	11.02

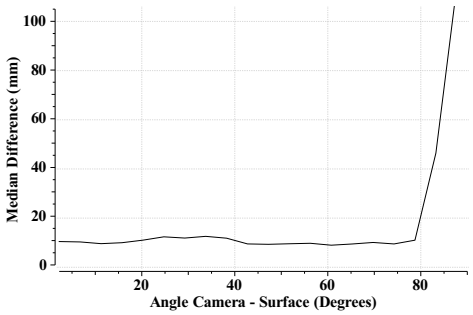
Table 6.2.: Median difference between 3D measurements and reconstructed 3D model (values in mm).

Even though the 3D reconstruction matches the measurements better than the ground truth 3D model, the 3D measurements differ from the reconstructed 3D model for three reasons. First, without 3D reconstruction, the 3D measurements vary due to random measurement noise. This effect is reduced by the 3D reconstruction (see last column of Table 6.2). Second, the systematic measurement errors depend on several factors, such as the distance of the camera to the surface and the measurement angle between the camera and the object surface. Thus, the single 3D measurements differ from the final 3D reconstruction, which models the average measurements of the depth camera. Third, the Kinect acquires discretized depth values: the depth is discretized into 11 bit. Thus, only 2048 different depth values are provided by the Kinect. At 1 m meausuring distance, the discretization is about 2 mm and at 2.5 m distance, all depth values from an interval of 25 mm are represented by the same depth value [KE12].

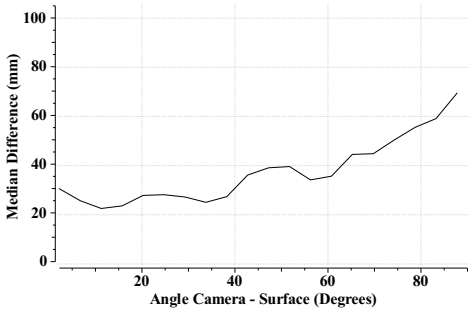
For these reasons, although they are smaller than for the single 3D measurements, there are differences as well between the difference detection with on-the-fly 3D reconstruction and the 3D model which is constructed from all 3D these measurements. During the 3D reconstruction, the reconstructed shape successively converges to the final 3D reconstruction shape. Thus, the differences become smaller for each successive frame.



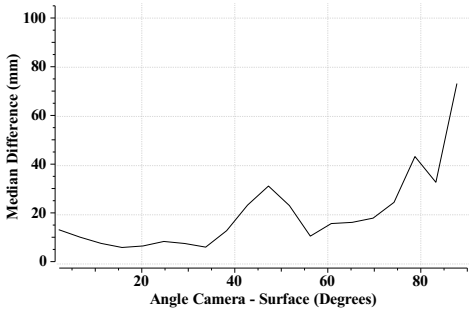
(a) SwissRanger 4000, CMM pose estimation, curved surface.



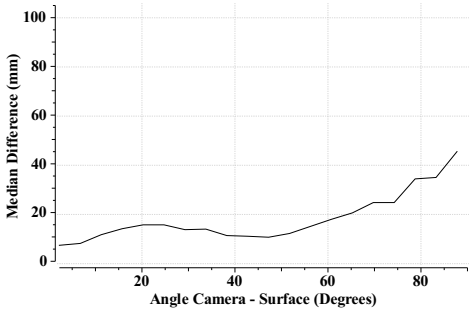
(b) Kinect, CMM pose estimation, curved surface.



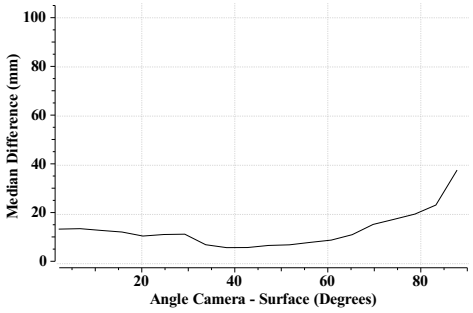
(c) SwissRanger 4000, CMM pose estimation, metallic object.



(d) Kinect, CMM pose estimation, metallic object.



(e) SwissRanger 4000, CMM pose estimation, white convex objects.



(f) Kinect, CMM pose estimation, white convex objects.

Figure 6.12.: Influence of the measurement angle between the camera and the object surface on the accuracy.

6.2.5. Influence of the angle on the measurement accuracy

To quantify the effect of the measurement angle on the measurement accuracy, the accuracy of the 3D difference detection was evaluated as a function of the measurement angle. First, the surface normal of the object surface was calculated for each pixel of the depth buffer image acquired from the rendered 3D model. Then, the angle between the surface normal and the view ray from the optical center of the depth camera through the current pixel was calculated. This angle varies between 0° and 90°.

Figure 6.12 visualizes the accuracy of the difference detection as a function of the measurement angle. The influence of the measurement angle is plotted for three different setups (the evaluation objects from Figure 6.9a, Figure 6.9b and the white convex objects from Figure 6.9c). For both cameras, the measurement accuracy decreases for large angles of more than about 60°.

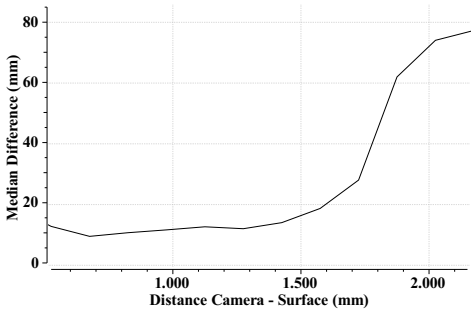
6.2.6. Influence of surface properties on the measurement accuracy

In industrial applications, the surfaces of objects often are metallic or have dark colors. Such surfaces are more difficult to measure with depth cameras than diffuse, light surfaces. To quantify the effect of different surface properties on the accuracy of the 3D difference detection, The 3D difference detection accuracy was evaluated for three objects with different surface properties. The accuracy was evaluated for convex shapes with a white respectively black surface (Figure 6.9c) and for a metallic, industrial object (Figure 6.9a).

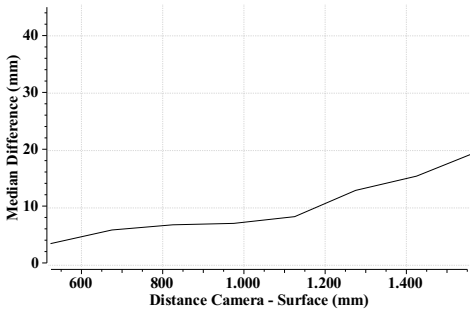
Figure 6.13 visualizes the results of this evaluation. For the SwissRanger depth camera, up to a distance of one meter, the accuracy of the difference detection is similar for the white and black shapes. However, for distances larger than one meter, the accuracy decreases faster for the black surfaces: the black surface partially absorbs the light emitted by the time-of-flight depth camera. Thus, less of the emitted and reflected light can be captured by the sensor of the time-of-flight camera, which decreases its measurement accuracy.

Even for close distances of the camera to the object, the measurement accuracy of the SwissRanger depth camera is low for the metallic object. This effect can be explained by the measurement principle of the time-of-flight depth camera: The light emitted by the time-of-flight camera is reflected multiple times by the metallic surface before it gets reflected to the depth camera. This increases the time it takes until the light emitted by the camera is captured by the camera sensor. Due to this prolonged time-of-flight of the emitted light, the depth camera overestimates the distance to the captured object.

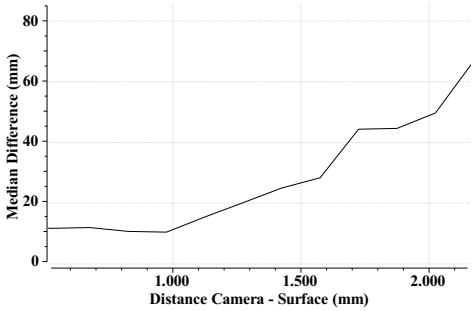
In contrast to the time-of-flight depth camera, the accuracy of the structured light Kinect depth camera gets less affected by the metallic surface of the industrial object. Although the surface of this object is very specular, for close distances of the Kinect depth camera to the surface, the distances to the object are measured with a high precision. Just as for the other evaluation objects, the measurement accuracy of the Kinect depends much more on the distance of the camera to the objects than on the surface properties of the captured objects. Figure 6.14 shows the difference visualization of metallic and black surfaces captured with a Kinect and a SwissRanger depth camera (see also Figure 5.6 for comparison



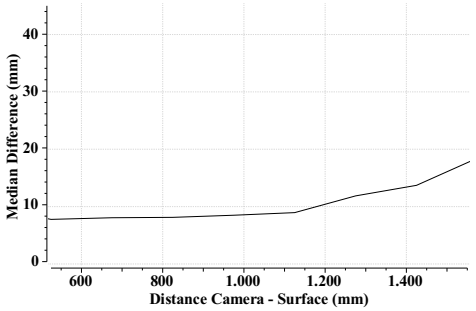
(a) SwissRanger 4000, CMM pose estimation, white convex objects.



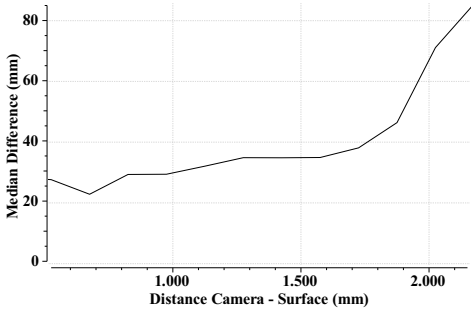
(b) Kinect, CMM pose estimation, white convex objects.



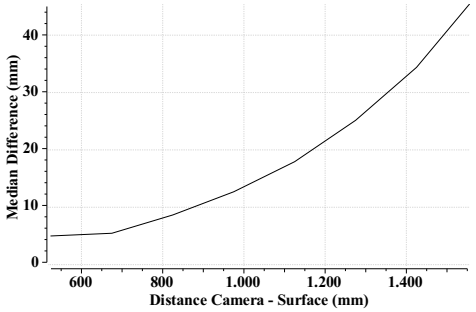
(c) SwissRanger 4000, CMM pose estimation, black convex objects.



(d) Kinect, CMM pose estimation, black convex objects.



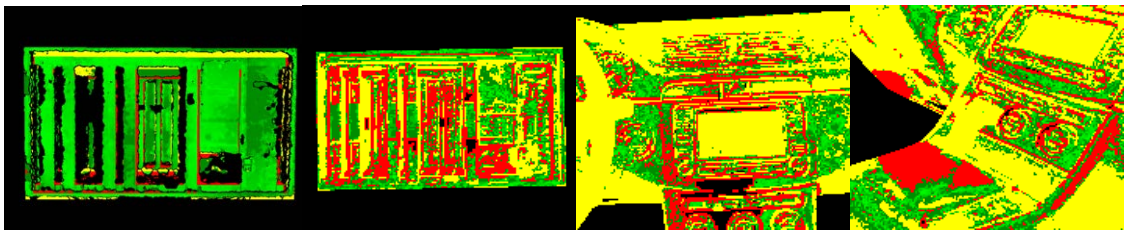
(e) SwissRanger 4000, CMM pose estimation, metallic object.



(f) Kinect, CMM pose estimation, metallic object.

Figure 6.13.: Influence of object surfaces (white, black and specular metallic) on the depth measurement accuracy.

with the depth image acquired by the Kinect). For such surfaces, the measurements acquired by the SwissRanger depth camera are much noisier than those acquired by the Kinect.



(a) Kinect (b) SwissRanger 4000 (c) SwissRanger 4000 (d) SwissRanger 4000

Figure 6.14.: Objects with metallic and black surfaces captured with a SwissRanger 4000 and a Kinect depth camera (same objects as in Figure 6.9a). The color encoding of the differences is the same as in Figure 5.6.

7. Concluding Remarks

This thesis introduced real time depth image based 3D difference detection. The proposed approach is the first solution with which it is possible to detect dense 3D differences in real time for a moving camera position. Previous approaches were restricted to static viewing positions, not real time capable or only provided a visual augmentation of 2D images with a 3D model, without actually measuring the 3D shape of the real object.

This thesis first provided an answer to the following research question:

Q1 ***How can 3D differences be detected in real time and from arbitrary viewpoints using a single depth camera?***

The answer to this question, provided by this thesis, is based on the confluence of computer vision and computer graphics. First, with computer vision, the pose of the depth camera relative to the 3D model is estimated in real time. Then, a computer graphics based analysis-by-synthesis approach provides a real-time mapping of the 3D measurements to corresponding 3D points on the surface of the 3D model. This combination of computer vision and computer graphics based analysis-by-synthesis is the key to real-time 3D difference detection, as it provides a 3D-3D mapping very efficiently: with this approach, the differences between a depth image with 600.000 depth measurements and a 3D model with 2.5 million triangles can be calculated in less than 15 milliseconds.

The second research question addressed approaches for enhancing the precision of the proposed difference detection:

Q2 ***Extending the first question, how can 3D differences be detected with a high precision?***

The accuracy of 3D difference detection is mainly limited by two factors: first, by inaccuracies of the pose estimation of the depth camera in relation to the 3D model. Second, by random noise and by systematic measurement errors of the depth measurements acquired by the depth camera. Therefore, this thesis discussed approaches for precise camera pose estimation and for reducing the measurement inaccuracies of depth images.

Based on this analysis, the combination of a depth camera with a coordinate measuring machine was proposed for precise pose estimation. Therefore, image based and 3D measurement based algorithms for estimating the hand-eye calibration between the depth camera and the coordinate measuring machine were introduced and evaluated comparatively. The precise pose estimation of the depth camera provided by the combination of the depth camera with the coordinate measuring machine is the first

part of the answer to Q2. The second part of the answer is provided by an improvement of the depth measurement accuracy.

In order to reduce the measurement inaccuracies of depth measurements captured by depth cameras, a 3D reconstruction algorithm was integrated in the 3D difference detection. This algorithm reconstructs a 3D model from the captured depth images. The surface of the captured scene is reconstructed in real time, while the depth camera is moved in order to detect 3D differences. Thus, gaps are filled from depth information acquired in other depth images. In addition, the fusion of 3D information from several depth images reduces the measurement noise.

The combination of precise depth imaging with real-time 3D difference detection, as proposed in this thesis, provided an answer to Q2. Finally, the third research question addressed the quantitative evaluation of the 3D difference detection accuracy:

Q3 ***Which accuracy can be achieved with concrete setups of the proposed concept for real time, depth image based 3D difference detection?***

This question was answered by the quantitative evaluation of the accuracy of different setups of the proposed 3D difference detection. On the one hand, a simulation quantified the influence of the accuracy of the intrinsic parameters, the accuracy of the pose estimation, the influence of noise in the captured depth measurements and the effects of approximated 3D meshes on the overall accuracy.

Furthermore, the accuracy was evaluated quantitatively with depth image sequences captured by a time-of-flight depth camera (SwissRanger 4000) and a structured light depth camera (Kinect). The accuracy was evaluated both for the basic setup (image based pose estimation, without 3D reconstruction) and for the proposed variant for precise 3D difference detection (pose estimation by combining the depth camera with a coordinate measuring machine, with 3D reconstruction).

With the basic setup and the structured light depth camera, differences of 8 to 24 millimeters can be detected from one meter measurement distance. With the enhancements proposed for precise 3D difference detection, finer details can be compared: with the proposed enhancements, differences of 4 to 12 millimeters can be detected from one meter measurement distance.

Conclusion By solving the challenges described by the three research questions, this thesis provides a solution for precise real-time 3D difference detection based on depth images. With the approach proposed in this thesis, dense 3D differences can be detected **in real time** and from arbitrary viewpoints using a single depth camera. Furthermore, by coupling the depth camera with a coordinate measuring machine and by integrating 3D reconstruction in the 3D difference detection, 3D differences can be detected with a **high precision**. As shown by the **quantitative evaluation**, differences of 4 to 12 millimeters can be detected with the proposed approach for precise real-time 3D difference detection based on depth images.

Future Work

This thesis described dense real-time 3D difference detection for static 3D models. Due to the limited operating range of depth cameras and coordinate measuring machines, it is best suited for detecting differences of objects which have a size of up to four meters. In future work, new approaches could be researched for 3D difference detection with non-rigid or parametrizable 3D models. Furthermore, the 3D difference detection could be extended to large scale 3D difference detection.

Large scale 3D difference detection The approach proposed in this work could be extended to large scale 3D difference detection by algorithms which combine the depth image based difference detection with difference detection using laser scan data. First, a laser scanner could be used to acquire a large reference 3D point cloud of the environment beyond the measurement range of the depth camera. Then, local 3D difference detection with a depth camera (as described in this thesis) could complement the global reference 3D point cloud. For example, with the depth image based 3D difference detection, differences between a 3D model and the real object could be detected at parts of the scanned object which are not visible from the point of view of the laser scanner (for example due to occlusions).

Parametrizable 3D models The problem definition stated in Section 1.2 defines that the configuration of the 3D model, which defines its 3D shape, needs to be known. However, the concepts and methods described in this thesis could also be used to find such a configuration. Consider the following problem formulation:

Given a real object and a parametrizable 3D model of this object. Find a configuration of the parameters of the 3D model such that the shape of the 3D model matches the shape of the real object as well as possible.

The methods described in this thesis could help to solve this problem formulation as well. Given a hypothesis about a possible configuration of the 3D model, the methods described in this thesis can be used to evaluate how well the shape of the 3D model matches the real object with the hypothesized parametrization. To get an estimation of the shape similarity, the 3D model and the real object first need to be aligned in an offline step as described in Section 3.2.1. Then, a depth image of the real object can be taken. This depth image can be compared with artificial depth images created by rendering the 3D model with different parametrizations. Thus, for the current point of view, the similarity of the real object and the 3D model can be calculated for different configurations of the 3D model. Such a difference estimation could be used to select a parametrization of the 3D model such that the shape of the 3D model matches the real object as well as possible.

Extending the approaches described in this thesis to non-rigid or parametrizable 3D models introduces two new main questions for future work. First, how can the parametrization be altered such that a suitable configuration is found efficiently? Second, can a suitable parameter configuration be found in real time? Taking this idea one step further could introduce new fields of research related to real-time

3D difference detection between the simulation of an articulated 3D model and the movements of a real object.

A. Publications

This thesis is partially based on the following publications:

1. KAHN, S.; HAUMANN, D.; WILLERT, V.: Hand-eye calibration with a depth camera: 2D or 3D? In: 9th International Conference on Computer Vision Theory and Applications (VISAPP) (2014), vol.3, pp. 481-489 [KHW14]
2. KAHN, S.; BOCKHOLT, U.; KUIJPER, A.; FELLNER, D. W.: Towards precise real-time 3D difference detection for industrial applications. In: Computers in Industry vol 64, nr 9 (2013). Elsevier Journal, pp. 1115-1128 [KBKF13]
3. KAHN, S.: Reducing the gap between augmented reality and 3D modeling with real-time depth imaging. In: Virtual Reality vol 17, nr 2 (2013). Springer Journal, pp. 111-123 [Kah13]
4. OLBRICH, M., GRAF, H., KAHN, S., ENGELKE, T., KEIL, J., RIESS, P., WEBEL, S., BOCKHOLT, U., PICINBONO, G.: Augmented reality supporting user-centric building information management. In: The Visual Computer vol 29, nr 10 (2013). Springer Journal, pp. 1093-1105 [OGK*13]
5. KAHN, S.; KUIJPER, A.: Fusing real-time depth imaging with high precision pose estimation by a measurement arm. In: International Conference on Cyberworlds (CW) (2012), pp. 256-260 [KK12]
6. KAHN, S.; OLBRICH, M.; ENGELKE, T.; KEIL, J.; RIESS, P.; WEBEL, S.; GRAF, H.; BOCKHOLT, U.; PICINBONO, G.: Beyond 3D "as-built" information using mobile AR enhancing the building lifecycle management. In: International Conference on Cyberworlds (CW) (2012), pp. 29-36 [KOE*12]
7. KAHN, S.; BOCKHOLT, U.: 3D Soll-Ist Abgleich mit Hilfe von Tiefenkameras. In: IFF-Wissenschaftstage 2012. Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme (2012), pp. 195-198 [KB11]
8. GRAF, H.; HAZKE, L.; KAHN, S.; MALERCZYK, C.: Accelerated real-time reconstruction of 3D deformable objects from multi-view video channels. In: 14th International Conference on Human-Computer Interaction (HCI) (2011), LNCS 6777, pp. 282–291 [GHKM11]
9. FRANKE, T.; KAHN, S.; JUNG, Y.; OLBRICH, M.: Enhancing realism of mixed reality applications through real-time depth imaging devices in X3D. In: ACM International Conference on 3D Web Technology (Web3D) (2011), pp. 71–79 [FKOJ11]

10. KAHN S., WUEST H., STRICKER D., FELLNER D. W.: 3D discrepancy check via augmented reality. In: International Symposium on Mixed and Augmented Reality (ISMAR) (2010), pp. 241–242 [KWSF10]
11. KAHN S., WUEST H., FELLNER D. W.: Time-of-flight based scene reconstruction with a mesh processing tool for model based camera tracking. In: 5th International Conference on Computer Vision Theory and Applications (VISAPP) (2010), vol. 1, pp. 302–309 [KWF10]
12. KAHN S., BOCKHOLT U.: 3D-Rekonstruktion mit einer Tiefenkamera für industrielle Augmented Reality Anwendungen. In 12. IFF-Wissenschaftstage 2009. Tagungsband: Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme (2009), pp. 105–112 [KB09]

Further publications:

1. KAHN, S.; KEIL, J.; MUELLER, B.; BOCKHOLT, U., FELLNER D. W.: Capturing of contemporary dance for preservation and presentation of choreographies in online scores. In: Digital Heritage (2013), pp. 273-280 [KKM*13]
2. KAHN, S.; KEIL, J.; ZOELLNER, M.; MUELLER, B.: Towards an affordable markerless acquisition of intangible contemporary dance choreographies at large-scaled stages. In: 13th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST) (2012), pp. 33–36 [KKZM12]
3. HARTMANN, P.; KAHN, S.; BOCKHOLT, U.; KUIJPER, A.: Towards symmetry axis based markerless motion capture. In: 8th Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS) (2011), pp. 73–82 [HK*11]
4. KAHN, S.; MALERCZYK, C.; GRAF, H.; BOCKHOLT, U.: Capturing motion skills with silhouette-based numerical pose estimation. In: International Conference on Multimodal Interfaces for Skills Transfer (SKILLS) (2011), pp. 1–4 [KMGB11]
5. KAHN S., KLUG T., FLENTGE F.: Modeling temporal dependencies between observed activities. In: International Conference on Multimodal Interfaces (ICMI). Proceedings of the 2007 workshop on tagging, mining and retrieval of human related activity information (2007), pp. 27–34 [KKF07]

B. Supervising Activities

The following list summarizes the student bachelor, diploma and master thesis supervised by the author. The results of these works were partially used as an input into the thesis.

B.1. Diploma and Master Theses

1. HARTMANN, P.: Tiefenbild basierte markerlose Erfassung menschlicher Bewegungen anhand von medialen Achsen. Master thesis, TU Darmstadt, 2011. [Har11]
2. WUNSCH R.: Modellbasierte Initialisierung der Kamerapose anhand des "Iterative Closest Point"-Algorithmus. Diploma thesis, FH Offenburg, 2010. [Wun10]
3. HAZKE L.: Beschleunigung der Berechnung einer visuellen Hülle mittels CUDA. Diploma thesis, FH Gießen-Friedberg, 2010. [Haz10]

B.2. Bachelor Theses

1. FRITSCH, V.: Motion Capturing durch zwei kombinierte Tiefenbilder. Bachelor thesis, Hochschule Darmstadt, 2012. [Fri12]
2. PELZER, M.: Vergleichende Evaluation der Rekonstruktionsgenauigkeit von Structure-from-Motion und Tiefenkameras anhand eines Messarmes. Bachelor thesis, TU Darmstadt, 2011. [Pel11]
3. BRIEMANN D.: Bildverbesserung von Time-of-Flight Bildern mit Hilfe von Markov Random Fields und 2D-Farbbildern. Bachelor thesis, Hochschule Darmstadt, 2010. [Bri10]
4. THÖNER M.: Vergleichende Evaluierung von Time-of-Flight und Structure from Motion Rekonstruktion zur Entwicklung eines kombinierten Kameratracking-Verfahrens. Bachelor thesis, TU Darmstadt, 2010. [Thö10]
5. KLEINE W.: Extraktion eines 3D-Kantenmodells aus Tiefenbildern. Bachelor thesis, TU Darmstadt, 2009. [Kle09]

Bibliography

[ABG*06] AKINCI B., BOUKAMP F., GORDON C., HUBER D., LYONS C., PARK K.: A formalism for utilization of sensor systems and integrated project models for active construction quality control. *Automation in Construction 15*, 2 (2006), 124 – 138. v, 31, 32, 33, 34

[Ald79] ALDRICH R. C.: *Remote Sensing of Wildland Resources: A State-of-the-Art Review*. Tech. rep., General Technical Report RM 71. Fort Collins, GIS World Inc., 1979. 26

[AML07] ANDREASSON H., MAGNUSSON M., LILIENTHAL A. J.: Has something changed here? autonomous difference detection for security patrol robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2007), pp. 3429–3435. v, 30, 33, 34

[ASW09] ALTMANNINGER K., SEIDL M., WIMMER M.: A survey on model versioning approaches. *International Journal of Web Information Systems 5* (2009), 271–304. 29

[ATAH13] ANIL E. B., TANG P., AKINCI B., HUBER D.: Deviation analysis method for the assessment of the quality of the as-is building information models generated from point cloud data. *Automation in Construction*, 0 (2013), –. 32, 34

[Azu97] AZUMA R. T.: A survey of augmented reality. *Presence: Teleoperators and Virtual Environments 6* (1997), 355–385. 54

[BBGB*12] BELHEDI A., BOURGEOIS S., GAY-BELLILE V., SAYD P., BARTOLI A., HAMROUNI K.: Non-parametric depth calibration of a tof camera. In *ICIP 2012* (2012), pp. 549–552. 23, 80, 81

[BBP*07] BECKER M., BLESER G., PAGANI A., STRICKER D., WUEST H.: An architecture for prototyping and application development of visual tracking systems. In *Capture, Transmission and Display of 3D Video (Proceedings of 3DTV-CON 07 [CD-ROM])* (2007). 68

[BBS07] BLESER G., BECKER M., STRICKER D.: Real-time vision-based tracking and reconstruction. *J. Real Time Image Proc. 2* (2007), 161–175. 17, 60, 83

[BCRM12] BRIÈRE-CÔTÉ A., RIVEST L., MARANZANA R.: Comparing 3d cad models: Uses, methods, tools and perspectives. *Computer-Aided Design & Applications 9*, 6 (2012), 771–794. 29

[BCRM13] BRIÈRE-CÔTÉ A., RIVEST L., MARANZANA R.: 3d cad model comparison: An evaluation of model difference identification technologies. *Computer-Aided Design & Applications 10*, 2 (2013), 173–195. 29

[BI12] BEUMIER C., IDRISSA M.: Building change detection from uniform regions. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Alvarez L., Mejail M., Gomez L., Jacobo J., (Eds.), vol. 7441 of *LNCS*. Springer Berlin Heidelberg, 2012, pp. 648–655. 26

[BM92] BESL P., MCKAY N.: A method for registration of 3-d shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence 1992* (1992), vol. 14(2), pp. 239–256. 15, 40, 61, 69, 83, 85, 101

[BOG*04] BLANC N., OGGIER T., GRUENER G., WEINGARTEN J., CODOUREY A., SEITZ P.: Miniaturized smart cameras for 3d-imaging in real-time. In *Sensors, 2004. Proceedings of IEEE* (2004), vol. 1, pp. 471–474. 18, 19, 20

[BOL05] BUTTGEN B., OGGIER T., LEHMANN M.: Ccd/cmos lock-in pixel for range imaging: Challenges, limitations and state-of-the-art. In *Proceedings of 1st Range Imaging Research Day, Zurich* (2005), pp. 21–32. 19

[Bos08] BOSCHÉ F.: *Automated Recognition of 3D CAD Model Objects in Dense Laser Range Point Clouds*. PhD thesis, University of Waterloo, 2008. v, 3, 32, 33, 34

[Bos10] BOSCHÉ F.: Automated recognition of 3d cad model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Elsevier Journal of Advanced Engineering Informatics 24*, 1 (2010), 107–118. v, 3, 32, 33, 34

[Bri10] BRIEMANN D.: *Bildverbesserung von Time-of-Flight Bildern mit Hilfe von Markov Random Fields und 2D-Farbbildern*. Bachelor thesis, Hochschule Darmstadt, 2010. 82, 119

[BS08] BUTTGEN B., SEITZ P.: Robust optical time-of-flight range imaging based on smart pixel structures. *IEEE Transactions on Circuits and Systems 55*, 6 (2008), 1512–1525. 18, 19

[BTHC06] BOSCHÉ F., TEIZER J., HAAS C. T., CALDAS C. H.: Integrating data from 3d cad and 3d cameras for real-time modeling. In *Proceedings of Joint International Conference on Computing and Decision Making in Civil and Building Engineering 2006* (2006), pp. 37–46. v, 31, 33

[BW10] BI Z., WANG L.: Advances in 3d data acquisition and processing for industrial applications. *Robotics and Computer-Integrated Manufacturing 26*, 5 (2010), 403 – 413. 16

[BWS06] BLESER G., WUEST H., STRICKER D.: Online camera pose estimation in partially known and dynamic scenes. In *ISMAR 2006: Proceedings of the Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality* (2006), pp. 56–65.

50

[Cal11] CALIBRATIONTOOLBOX G. C.: Gml camera calibration toolbox, 2011. http://graphics.cs.msu.ru/en/science/research/calibration/cpp. 39

[CALT12] CHOW J. C. K., ANG K. D., LICHTI D. D., TESKEY W. F.: Performance analysis of a low-cost triangulation-based 3d camera: Microsoft kinect system. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIX-B5* (2012), 175–180. 24, 80

[CL90] CHAO T.-H., LIU H.-K.: Real-time image difference detection using a polarization rotation spacial light modulator. Patent, 03 1990. US 4908702. 25

[CS11] CUI Y., STRICKER D.: 3d shape scanning with a kinect. In *ACM SIGGRAPH 2011 Posters* (2011), pp. 57–57. 83

[CSD*10] CUI Y., SCHUON S., DEREK C., THRUN S., THEOBALT C.: 3d shape scanning with a time-of-flight camera. In *Proc. of IEEE CVPR 2010* (2010), pp. 1173–1180. 83

[CSK05] CHETVERIKOV D., STEPANOV D., KRSEK P.: Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing 23*, 3 (2005), 299 – 309. 15, 40, 62

[DLG*12] DU P., LIU S., GAMBA P., TAN K., XIA J.: Fusion of difference images for change detection over urban areas. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of 5*, 4 (2012), 1076–1086. 26

[DRMS07] DAVISON A., REID I., MOLTON N., STASSE O.: Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 6 (2007), 1052–1067. 83

[DS12] DOBOŠ J., STEED A.: 3d diff: an interactive approach to mesh differencing and conflict resolution. In *SIGGRAPH Asia 2012 Technical Briefs* (2012), pp. 20:1–20:4. 29

[DT05] DIEBEL J., THRUN S.: An application of markov random fields to range sensing. In *NIPS'05* (2005), pp. 291–298. 82

[DWJM98] DORAI C., WANG G., JAIN A., MERCER C.: Registration and integration of multiple object views for 3d model construction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20*, 1 (1998), 83–89. 15

[EOHM09] EDELER T., OHLIGER K., HUSSMANN S., MERTINS A.: Super resolution of time-of-flight depth images under consideration of spatially varying noise variance. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (2009), pp. 1185 –1188. 82

[EW76] EBERSOLE J. F., WYANT J. C.: Real-time optical subtraction of photographic imagery for difference detection. *Applied Optics 15*, 4 (1976), 871–876. 25

[Far13a] Faro Focus3D, 2013. http://www.faro.com/focus. Date of access: 02/2013. 2, 16

[Far13b] Faro ScanArm, 2013. http://measuring-arms.faro.com/scanarm. Date of access: 02/2013. 2, 16

[FAT11] FOIX S., ALENYA G., TORRAS C.: Lock-in time-of-flight (tof) cameras: A survey. *Sensors Journal, IEEE 11*, 9 (2011), 1917–1926. 23, 81

[FG11] FITE-GEORGEL P.: *Augmented Reality Tools for Digital Plant Engineering*. PhD thesis, TU Munich, 2011. v, 3, 26, 27, 33, 34

[FH08] FUCHS S., HIRZINGER G.: Extrinsic and depth calibration of tof-cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008* (2008), pp. 1–6. 23, 80, 81

[Fit03] FITZGIBBON A. W.: Robust registration of 2d and 3d point sets. *Image and Vision Computing 21*, 13-14 (2003), 1145 – 1153. 15

[FKOJ11] FRANKE T., KAHN S., OLBRICH M., JUNG Y.: Enhancing realism of mixed reality applications through real-time depth imaging devices in x3d. In *Proceedings of the 16th ACM International Conference on 3D Web Technology* (New York, NY, USA, 2011), Web3D 2011, ACM, pp. 71–79. 2, 35, 54, 117

[FP02] FORSYTH D. A., PONCE J.: *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002. 9, 17

[FREM04] FARSIU S., ROBINSON M., ELAD M., MILANFAR P.: Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing 13*, 10 (oct. 2004), 1327 –1344. 81

[Fri12] FRITSCH V.: *Motion Capturing durch zwei kombinierte Tiefenbilder*. Bachelor thesis, Hochschule Darmstadt, 2012. 119

[FSA10] FREEDMAN B., SHPUNT A., ARIELI Y.: Distance-varying illumination and imaging techniques for depth mapping. Patent Application, 11 2010. US 2010/0290698 A1. 21, 22

[Fuc12] FUCHS S.: *Calibration and Multipath Mitigation for Increased Accuracy of Time-of-Flight Camera Measurements in Robotic Applications*. PhD thesis, TU Berlin, Germany, 2012. 66

[GBSN09] GEORGEL P., BENHIMANE S., SOTKE J., NAVAB N.: Photo-based industrial augmented reality application using a single keyframe registration procedure. In *ISMAR 2009: Proceedings of the 8th IEEE and ACM International Symposium on Mixed and Augmented Reality* (2009), pp. 187–188. v, 3, 26, 27, 33, 34

[GGV*10] GRUBER L., GAUGLITZ S., VENTURA J., ZOLLMANN S., HUBER M., SCHLEGEL M., KLINKER G., SCHMALSTIEG D., HÖLLERER T.: The city of sights: Design, construction, and measurement of an augmented reality stage set. In *Proc. Nineth IEEE International Symposium on Mixed and Augmented Reality (ISMAR'10)* (Seoul, Korea, Oct. 13-16 2010), pp. 157–163. 62

[GHKM11] GRAF H., HAZKE L., KAHN S., MALERCZYK C.: Accelerated real-time reconstruction of 3d deformable objects from multi-view video channels. In *Digital Human Modeling*, vol. 6777 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 282–291. 117

[GJRVF*13] GONZALEZ-JORGE H., RIVEIRO B., VAZQUEZ-FERNANDEZ E., MARTINEZ-SANCHEZ J., ARIAS P.: Metrological evaluation of microsoft kinect and asus xtion sensors. *J. Measurement*, 0 (2013), http://dx.doi.org/10.1016/j.measurement.2013.01.011. 20

[GMZC09] GARRO V., MUTTO C. D., ZANUTTIGH P., CORTELAZZO G. M.: A novel interpolation scheme for range data with side information. In *Proceedings of the 2009 Conference for Visual Media Production* (2009), CVMP '09, pp. 52–60. 82

[GRS06] GALL J., ROSENHAHN B., SEIDEL H.-P.: Robust pose estimation with 3D textured models. In *Pacific-Rim Symposium on Image and Video Technology (PSIVT)* (2006), pp. 84–95. 54

[GSB*07] GEORGEL P., SCHROEDER P., BENHIMANE S., HINTERSTOISSER S., APPEL M., NAVAB N.: An industrial augmented reality solution for discrepancy check. In *ISMAR 2007: Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007), pp. 1–4. v, 26, 27, 33, 34

[GSN09] GEORGEL P., SCHROEDER P., NAVAB N.: Navigation tools for viewing augmented cad models. *IEEE Computer Graphics and Applications 29*, 6 (2009), 65–73. 27

[Hal04] HALLER M.: Photorealism or/and non-photorealism in augmented reality. In *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry* (2004), VRCAI '04, pp. 189–196. 54

[Har11] HARTMANN P.: *Tiefenbild basierte markerlose Erfassung menschlicher Bewegungen anhand von medialen Achsen*. Master thesis, TU Darmstadt, 2011. 119

[Haz10] HAZKE L.: *Beschleunigung der Berechnung einer visuellen Hülle mittels CUDA*. Diploma thesis, FH GieSSen-Friedberg, 2010. 119

[HK*11] HARTMANN P., KAHN S., , BOCKHOLT U., KUIJPER A.: Towards symmetry axis based markerless motion capture. In *8th Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS)* (2011), pp. 73–82. 118

[HLCH12] HANSARD M., LEE S., CHOI O., HORAUD RADU P.: *Time of Flight Cameras: Principles, Methods, and Applications*. SpringerBriefs in Computer Science. Springer, 2012. 18

[HRH08] HUSSMANN S., RINGBECK T., HAGEBEUKER B.: A performance review of 3d tof vision systems in comparison to stereo vision systems. In *Stereo Vision*, Bhatti A., (Ed.). InTech, 2008. 18

[HS97] HEIKKILA J., SILVEN O.: A four-step camera calibration procedure with implicit image correction. In *IEEE Conference on Computer Vision and Pattern Recognition*

(1997), pp. 1106–1112. 11

[IKH*11] IZADI S., KIM D., HILLIGES O., MOLYNEAUX D., NEWCOMBE R., KOHLI P., SHOTTON J., HODGES S., FREEMAN D., DAVISON A., FITZGIBBON A.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *24th ACM symposium on User interface software and technology (UIST)* (2011), pp. 559–568. 7, 83, 84

[JG11] JIN X., GOTO S.: Encoder adaptable difference detection for low power video compression in surveillance system. *Signal Processing: Image Communication 26*, 3 (2011), 130 – 142. 28

[Jon95] JONES M. W.: *3D distance from a point to a triangle*. Tech. rep., Department of Computer Science, University of Wales, 1995. 36

[JR11] JIOSSY R., ROSENBERG O.: Encoding video using scene change detection. Patent Application, 03 2011. US 2011/0051010 A1. 28

[Kah13] KAHN S.: Reducing the gap between augmented reality and 3d modeling with real-time depth imaging. *Virtual Reality 17*, 2 (2013), 111–123. 10.1007/s10055-011-0203-0. 2, 35, 42, 54, 55, 93, 117

[KAM06] KIL Y. J., AMENTA N., MEDEROS B.: Laser scanner super-resolution. In *Eurographics Symposium on Point-Based Graphics 2006* (2006), pp. 9–15. 82

[KB09] KAHN S., BOCKHOLT U.: 3D-Rekonstruktion mit einer Tiefenkamera für industrielle Augmented Reality Anwendungen. In *12. IFF-Wissenschaftstage 2009. Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme* (2009), pp. 105–112. 118

[KB11] KAHN S., BOCKHOLT U.: 3d soll-ist abgleich mit hilfe von tiefenkameras. In *15. IFF-Wissenschaftstage 2012. Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme* (2011), pp. 195–198. 117

[KBKF13] KAHN S., BOCKHOLT U., KUIJPER A., FELLNER D. W.: Towards precise real-time 3d difference detection for industrial applications. *Computers in Industry 64*, 9 (2013), 1115–1128. 35, 101, 117

[KBKL09] KOLB A., BARTH E., KOCH R., LARSEN R.: Time-of-flight sensors in computer graphics. In *Proc. Eurographics (State-of-the-Art Report)* (2009), pp. 119–134. 2, 18

[KE12] KHOSHELHAM K., ELBERINK S. O.: Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors 12*, 2 (2012), 1437–1454. 23, 24, 80, 107

[KH13] KIM D.-W., HA J.-E.: Hand/eye calibration using 3d-3d correspondences. *Applied Mechanics and Materials 319* (2013), 532–535. 66

[KHW14] KAHN S., HAUMANN D., WILLERT V.: Hand-eye calibration with a depth camera: 2d or 3d? In *9th International Conference on Computer Vision Theory and Applications (VISAPP)* (2014), vol. 3, pp. 481–489. 59, 66, 117

[KI08] KAHLMANN T., INGENSAND H.: Calibration and development for increased accuracy of 3d range imaging cameras. *Journal of Applied Geodesy 2*, 1 (2008), 1–11. 80, 81

[KK12] KAHN S., KUIJPER A.: Fusing real-time depth imaging with high precision pose estimation by a measurement arm. In *2012 International Conference on Cyberworlds (CW)* (2012), pp. 256–260. 35, 59, 66, 117

[KKF07] KAHN S., KLUG T., FLENTGE F.: Modeling temporal dependencies between observed activities. In *International Conference on Multimodal Interfaces (ICMI). Proceedings of the 2007 workshop on Tagging, mining and retrieval of human related activity information* (2007), ACM, pp. 27–34. 118

[KKM*13] KAHN S., KEIL J., MUELLER B., BOCKHOLT U., FELLNER D. W.: Capturing of contemporary dance for preservation and presentation of choreographies in online scores. In *Digital Heritage (2013)* (2013), pp. 273–280. 118

[KKZM12] KAHN S., KEIL J., ZOELLNER M., MUELLER B.: Towards an affordable markerless acquisition of intangible contemporary dance choreographies at large-scaled stages. In *13th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)* (2012), pp. 33–36. 118

[Kle09] KLEINE W.: *Extraktion eines 3D-Kantenmodells aus Tiefenbildern*. Bachelor thesis, TU Darmstadt, 2009. 119

[KM09] KLEIN G., MURRAY D.: Parallel tracking and mapping on a camera phone. In *Proceedings of the eigth IEEE and ACM international symposium on mixed and augmented reality (ISMAR'09)* (Orlando, October 2009), pp. 83–86. 83

[KMGB11] KAHN S., MALERCZYK C., GRAF H., BOCKHOLT U.: Capturing motion skills with silhouette-based numerical pose estimation. In *International Conference on Multimodal Interfaces for Skills Transfer (SKILLS)* (2011), pp. 1–4. 118

[KOE*12] KAHN S., OLBRICH M., ENGELKE T., KEIL J., RIESS P., WEBEL S., GRAF H., BOCKHOLT U., PICINBONO G.: Beyond 3d "as-built" information using mobile ar enhancing the building lifecycle management. In *2012 International Conference on Cyberworlds (CW)* (2012), pp. 29–36. 117

[KRI06] KAHLMANN T., REMONDINO H., INGENSAND H.: Calibration for increased accuracy of the range imaging camera swissranger. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 35*, 5 (2006), 136–141. 81

[KWF10] KAHN S., WUEST H., FELLNER D. W.: Time-of-flight based scene reconstruction with a mesh processing tool for model based camera tracking. In *5th International Conference on Computer Vision Theory and Applications (VISAPP)* (2010), vol. 1, pp. 302–309. 118

[KWSF10] KAHN S., WUEST H., STRICKER D., FELLNER D. W.: 3d discrepancy check via augmented reality. In *9th IEEE International Symposium on Mixed and Augmented*

Reality (ISMAR) 2010 (2010), pp. 241–242. 35, 101, 118

[KZ08] KIM Y. M., ZHU K.: *Super-resolution 3d multiview reconstruction using time-of-flight depth sensors*. Tech. rep., Stanford University, 2008. 82

[LBMN11] LIEBERKNECHT S., BENHIMANE S., MEIER P., NAVAB N.: Benchmarking template-based tracking algorithms. *Virtual Reality 15* (2011), 99–108. 62

[LC86] LIU H.-K., CHAO T.-H.: Optical image subtraction techniques, 1975 - 1985. *Hybrid Image Processing Proc. SPIE 0638*, 55 (1986), 55–65. 25

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 1987)* (1987), vol. 21, pp. 163–169. 83, 86

[Lei13] Leica laser scanner, 2013. http://hds.leica-geosystems.com. Date of access: 02/2013. 2, 16

[LF05] LEPETIT V., FUA P.: Monocular model-based 3D tracking of rigid objects: A survey. In *Foundations and Trends in Computer Graphics and Vision* (2005), vol. 1, pp. 1–89. 54

[LF06] LEPETIT V., FUA P.: Keypoint recognition using randomized trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 9 (2006), 1465–1479. 50

[LHL11] LANGMANN B., HARTMANN K., LOFFELD O.: Comparison of depth super-resolution methods for 2d/3d images. *International Journal of Computer Information Systems and Industrial Management Applications 3* (2011), 635 –645. 82

[LK06] LINDNER M., KOLB A.: Lateral and depth calibration of pmd-distance sensors. In *Advances in Visual Computing*, Bebis G., Boyle R., Parvin B., Koracin D., Remagnino P., Nefian A., Meenakshisundaram G., Pascucci V., Zara J., Molineros J., Theisel H., Malzbender T., (Eds.), vol. 4292 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 524–533. 81

[LKH07] LINDNER M., KOLB A., HARTMANN K.: Data-fusion of pmd-based distance-information and high-resolution rgb-images. In *Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS), Session on Algorithms for 3D TOF-cameras* (2007), vol. 1, pp. 121–124. 54

[LMPD11] LU J., MIN D., PAHWA R., DO M.: A revisit to mrf-based depth map super-resolution and enhancement. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2011), pp. 985 –988. 82

[LSKK10] LINDNER M., SCHILLER I., KOLB A., KOCH R.: Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding 114*, 12 (2010), 1318–1328. 23, 80, 81

[LSP*09] LEE C.-H., SONG K.-W., PARK Y.-O., KIM Y.-S., JOO Y.-H., PARK T.-S., KWON J.-H., JOUNG D.-Y., PARK J.-S., KIM S.-K., KIM Y.-G., OH Y.-J.: System and method for enhanced video communication using real-time scene-change

detection for control of moving-picture encoding data rate. Patent Application, 04 2009. US 2009/0097546 A1. 28

[Mac11] MACCORMICK J.: How does the kinect work? talk, 2011. http://users.dickinson.edu/ jmac/selected-talks/. Date of access: 05/2013. 22

[MAP*13] MALPICA J. A., ALONSO M. C., PAPI F., AROZARENA A., MARTINEZ DE AGIRRE A.: Change detection of buildings from satellite imagery and lidar data. *International Journal of Remote Sensing 34*, 5 (2013), 1652–1675. 29

[Mes09] MESAIMAGING: SR4000 user manual (version 2.0), 2009. 18, 71

[Mes13] Mesa Imaging, 2013. http://www.mesa-imaging.ch. Date of access: 02/2013. 23, 24, 101

[ND10] NEWCOMBE R., DAVISON A.: Live dense reconstruction with a single moving camera. In *IEEE conference on computer vision and pattern recognition (CVPR)* (2010), pp. 1498–1505. 17, 83, 84

[NDB*10] NUNEZ P., DREWS P., BANDERA A., ROCHA R., CAMPOS M., DIAS J.: Change detection in 3d environments based on gaussian mixture model and robust structural matching for autonomous robotic applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2010), pp. 2633–2638. v, 30, 33, 34

[NDR*09] NUNEZ P., DREWS P., ROCHA R., CAMPOS M., DIAS J.: Novelty detection and 3d shape retrieval based on gauusian mixture models for autonomous surveillance robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2009), pp. 4724–4730. 30

[NGL10] NAGESH P., GOWDA R., LI B.: Fast gpu implementation of large scale dictionary and sparse representation based vision problems. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on* (2010), pp. 1570 – 1573. 82

[NIH*11] NEWCOMBE R. A., IZADI S., HILLIGES O., MOLYNEAUX D., KIM D., DAVISON A. J., KOHLI P., SHOTTON J., HODGES S., FITZGIBBON A.: Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality* (2011), ISMAR 2011, pp. 127–136. xi, 7, 61, 62, 83, 84

[NLD11] NEWCOMBE R. A., LOVEGROVE S. J., DAVISON A. J.: Dtam: Dense tracking and mapping in real-time. *Computer Vision, IEEE International Conference on* (2011), 2320–2327. 83, 84

[Nof99] NOF S. Y.: *Handbook of Industrial Robotics*, 2nd ed. John Wiley & Sons, Inc., New York, NY, USA, 1999. 16, 62

[OGK*13] OLBRICH M., GRAF H., KAHN S., ENGELKE T., KEIL J., RIESS P., WEBEL S., BOCKHOLT U., PICINBONO G.: Augmented reality supporting user-centric building

information management. *The Visual Computer 29*, 10 (2013), 1093–1105. 117

[OLB06] OGGIER T., LUSTENBERGER F., BLANC N.: Miniature 3d tof camera for real-time imaging. In *Perception and Interactive Technologies* (2006), pp. 212–216. 2, 18

[OLK*04] OGGIER T., LEHMANN M., KAUFMANN R., SCHWEIZER M., RICHTER M., METZLER P., LANG G., LUSTENBERGER F., BLANC N.: An all-solid-state optical range camera for 3d real-time imaging with sub-centimeter depth resolution (swissranger). *Optical Design and Engineering* (2004), 534–545. 18

[Ope13] OPENNI: Openni framework, 2013. http://www.openni.org/. Date of access: 02/2013. 101

[Pel11] PELZER M.: *Vergleichende Evaluation der Rekonstruktionsgenauigkeit von Structure-from-Motion und Tiefenkameras anhand eines Messarmes*. Bachelor thesis, TU Darmstadt, 2011. 119

[PF06] PRADOS E., FAUGERAS O.: Shape from shading. In *Handbook of Mathematical Models in Computer Vision*, Paragios N., Chen Y., Faugeras O., (Eds.). Springer US, 2006, pp. 375–388. 16

[Pia11] PIATTI D.: *Time-of-Flight cameras: tests, calibration and multi-frame registration for automatic 3D object reconstruction*. PhD thesis, Politecnico di Torino, Italy, 2011. 69

[PMK06] PENTENRIEDER K., MEIER P., KLINKER G.: Analysis of tracking accuracy for single-camera square-marker-based tracking. In *Third Workshop on Virtual and Augmented Reality of the GI-Fachgruppe VR/AR* (2006), p. 4. 60

[PNF*08] POLLEFEYS M., NISTER D., FRAHM J.-M., AKBARZADEH A., MORDOHAI P., CLIPP B., ENGELS C., GALLUP D., KIM S.-J., MERRELL P., SALMI C., SINHA S., TALTON B., WANG L., YANG Q., STEWENIUS H., YANG R., WELCH G., TOWLES H.: Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision 78*, 2-3 (2008), 143–167. 83, 84

[PNPNGLFR05] PEREZ NAVA F., PEREZ NAVA A., GALVEZ LAMOLDA J. M., FUENTES REDONDO M.: Change detection for remote sensing images with graph cuts. *Image and Signal Processing for Remote Sensing XI Proc. SPIE 5982* (2005), 1–14. 26

[POPS09] PAOLINI A. L., ORTIZ F., PRICE D. K., SPAGNOLI K. E.: Development of a gpu-accelerated super resolution solver. In *Proc. SPIE 7348* (2009), pp. 1–11. 82

[Pri13] PrimeSense, 2013. http://www.primesense.com. Date of access: 02/2013. 20

[PSP09] PANAGOPOULOS A., SAMARAS D., PARAGIOS N.: Robust shadow and illumination estimation using a mixture model. In *IEEE conference on computer vision and pattern recognition (CVPR)* (2009), pp. 651 –658. 54

[Pul99] PULLI K.: Multiview registration for large data sets. In *3DIM* (1999), pp. 160–168. 83

[PZ09] PICKUP L. C., ZISSERMAN A.: Automatic retrieval of visual continuity errors in movies. In *Proceedings of the ACM International Conference on Image and Video Retrieval* (2009), pp. 7:1–7:8. 28

[RAAKR05] RADKE R. J., ANDRA S., AL-KOFAHI O., ROYSAM B.: Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing 14* (2005), 294–307. 26

[RC11] RUSU R. B., COUSINS S.: 3d is here: Point cloud library (pcl). In *International Conference on Robotics and Automation 2011* (2011), pp. 1–4. 15

[Ren02] RENSINK R. A.: Change detection. *Annual Review of Psychology*, 53 (2002), 245–277. 25

[RFHJ08] RAPP H., FRANK M., HAMPRECHT F., JAHNE B.: A theoretical and experimental investigation of the systematic errors and statistical uncertainties of time-of-flight-cameras. *International Journal of Intelligent Systems Technologies and Applications 5*, 3 (2008), 402–413. 23

[RHHL02] RUSINKIEWICZ S., HALL-HOLT O., LEVOY M.: Real-time 3D model acquisition. *ACM Transactions on Graphics 21*, 3 (2002). 83, 84

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. *In Proc. 3rd Intl. Conf. on 3-D Digital Imaging and Modeling* (2001), 224–231. 40, 62, 69, 83

[RRB12] REINBACHER C., RUTHER M., BISCHOF H.: Ronect: Hand mounted depth sensing using a commodity gaming sensor. In *21st International Conference on Pattern Recognition (ICPR)* (2012), pp. 461–464. 66

[SBK08] SCHILLER I., BEDER C., KOCH R.: Calibration of a pmd-camera using a planar calibration pattern together with a multi-camera setup. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2008), vol. XXI. ISPRS Congress, pp. 297–302. 40, 49, 80

[SBSS08] SWADZBA A., BEUTER N., SCHMIDT J., SAGERER G.: Tracking objects in 6d for reconstructing static scenes. In *Computer Vision and Pattern Recognition Workshop: Time of Flight Camera based Computer Vision* (2008), pp. 1–7. 81

[SGC10] STÜHMER J., GUMHOLD S., CREMERS D.: Real-time dense geometry from a hand-held camera. In *Proceedings of the 32nd DAGM conference on Pattern recognition* (2010), pp. 11–20. 83, 84

[SH06] STROBL K. H., HIRZINGER G.: Optimal hand-eye calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2006* (2006), pp. 4647–4653. 40, 65

[She64] SHEPARD J. R.: A concept of change detection. *Photogrammetrical Engineering 30* (1964), 648–651. 25, 26

[Sin89] SINGH A.: Review article digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing 10*, 6 (1989), 989–1003. 26

[SJP13] SMISEK J., JANCOSEK M., PAJDLA T.: 3d with kinect. In *Consumer Depth Cameras for Computer Vision*, Fossati A., Gall J., Grabner H., Ren X., Konolige K., (Eds.), Advances in Computer Vision and Pattern Recognition. Springer London, 2013, pp. 3–25. 49, 80

[SKK*10] SON H., KIM C., KIM H., HAN S., KIM M.: Trend analysis of research and development on automation and robotics technology in the construction industry. *KSCE Journal of Civil Engineering 14*, 2 (2010), 131–139. 16, 62

[SMAL13] STOYANOV T., MOJTAHEDZADEH R., ANDREASSON H., LILIENTHAL A. J.: Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications. *Robotics and Autonomous Systems 61*, 10 (2013), 1094–Ű1105. 69

[SS08] SCHOENFELDER R., SCHMALSTIEG D.: Augmented reality for industrial building acceptance. In *IEEE Virtual Reality (VR '08)* (2008), pp. 83–90. v, 26, 27, 33, 34

[ST94] SHI J., TOMASI C.: Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)* (1994), pp. 593–600. 60

[STD09] SANSONI G., TREBESCHI M., DOCCHIO F.: State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors 9*, 1 (2009), 568–601. 16

[STDM*13] STAL C., TACK F., DE MAEYER P., DE WULF A., GOOSSENS R.: Airborne photogrammetry and lidar for dsm extraction and 3d change detection over an urban area - a comparative study. *International Journal of Remote Sensing 34*, 4 (2013), 1087–1110. 29

[STDT09] SCHUON S., THEOBALT C., DAVIS J., THRUN S.: Lidarboost: Depth superresolution for tof 3d shape scanning. In *CVPR* (2009), IEEE, pp. 343–350. 82, 83

[SZ08] SHPUNT A., ZALEVSKY Z.: Depth-varying light fields for three dimensional sensing. Patent Application, 05 2008. US 2008/0106746 A1. 21, 22

[TAAH11] TANG P., ANIL E., AKINCI B., HUBER D.: Efficient and effective quality assessment of as-is building information models and 3d laser-scanned data. In *Proceedings of the ASCE International Workshop on Computing in Civil Engineering* (2011). v, 32, 33, 34

[TAH09] TANG P., AKINCI B., HUBER D.: Characterization of three algorithms for detecting surface flatness defects from dense point clouds. *Three-Dimensional Imaging Metrology Proc. SPIE 7239* (2009), 72390N–72390N–12. v, 31, 32, 33, 34

[Tan92] TANNOCK J.: Coordinate measuring machines. In *Automating Quality Systems*. Springer Netherlands, 1992, pp. 131–148. 16, 62

[Thö10] THÖNER M.: *Vergleichende Evaluierung von Time-of-Flight und Structure from Motion Rekonstruktion zur Entwicklung eines kombinierten Kameratracking-Verfahrens*. Bachelor thesis, TU Darmstadt, 2010. 119

[TL88] TSAI R. Y., LENZ R. K.: A new technique for fully autonomous and efficient 3d robotics hand-eye calibration. In *Proceedings of the 4th international symposium on Robotics Research 1998* (1988), pp. 287–297. 40, 65

[Tsa87] TSAI R.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of 3*, 4 (1987), 323–344. 60

[TV08] TANGELDER J. W., VELTKAMP R. C.: A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications 39*, 3 (2008), 441–471. 29

[Ume91] UMEYAMA S.: Least-squares estimation of transformation parameters between two point patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 13*, 4 (1991), 376–380. 15, 40, 49

[VDC12] VIEIRA A., DREWS P., CAMPOS M. F. M.: Efficient change detection in 3d environment for autonomous surveillance robots based on implicit volume. In *IEEE International Conference on Robotics and Automation (ICRA)* (2012), pp. 2999–3004. v, 30, 33, 34

[WBSW07] WEBEL S., BECKER M., STRICKER D., WUEST H.: Identifying differences between cad and physical mock-ups using ar. In *ISMAR 2007: Proceedings of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007), pp. 281–282. v, 31, 33, 34

[WRM*08] WAGNER D., REITMAYR G., MULLONI A., DRUMMOND T., SCHMALSTIEG D.: Pose tracking from natural features on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (2008), pp. 125–134. 60

[Wue08] WUEST H.: *Efficient Line and Patch Feature Characterization and Management for Real-time Camera Tracking*. PhD thesis, TU Darmstadt, 2008. 12, 13, 60

[Wun10] WUNSCH R.: *Modellbasierte Initialisierung der Kamerapose anhand des 'Iterative Closest Point'-Algorithmus*. Diploma thesis, FH Offenburg, 2010. 61, 119

[WWK11] WIENTAPPER F., WUEST H., KUIJPER A.: Composing the feature map retrieval process for robust and ready-to-use monocular tracking. *Computers & Graphics 35*, 4 (2011), 778 – 788. 17, 40, 48, 49, 50, 60, 61

[XXOEV12] XIAO W., XU S., OUDE ELBERINK S., VOSSELMAN G.: Change detection of trees in urban areas using multi-temporal airborne lidar point clouds. *Remote Sensing of the Ocean, Sea Ice, Coastal Waters, and Large Water Regions Proc. SPIE 8532* (2012), 853207–853207–10. 29

[YHY07] YUBIN Y., HUI L., YAO Z.: Content-based 3-d model retrieval: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37*, 6 (2007), 1081–1098. 29

[ZDB08] ZHOU F., DUH H. B.-L., BILLINGHURST M.: Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *ISMAR 2008: IEEE / ACM International Symposium on Mixed and Augmented Reality* (2008), vol. 0, pp. 193–202. 54

[Zha99] ZHANG Z.: Flexible camera calibration by viewing a plane from unknown orientations. In *IEEE International Conference on Computer Vision* (1999), vol. 1, pp. 666–673. 60

[ZSMG07] ZALEVSKY Z., SHPUNT A., MAIZELS A., GARCIA J.: Method and system for object reconstruction. Patent Application, 04 2007. WO 2007/043036 A1. 2, 20, 21