

Ein Framework zur Simulation optischer Sensoren

Steffen Sauer¹, Thomas Dunker¹, Michael Heizmann²

¹ Fraunhofer Institut für Fabrikbetrieb und –automatisierung IFF, Sandtorstraße 22, 39106 Magdeburg

² Karlsruher Institut für Technologie (KIT) Hertzstraße 16, 76187 Karlsruhe

Zusammenfassung

In dieser Arbeit wird ein neuartiges Framework vorgestellt, mit dem sich die Erzeugung von optischen Messdaten simulieren lässt. Im Vergleich zu bisherigen Arbeiten liegt hier der Fokus auf einer schnellen, realitätsnahen Approximation des Sensorverhaltens. Das Framework besteht aus Verfahren zur merkmalsbasierten Simulation von Kamerabildern, sowie zur 3D-Messdatensimulation von Laser-Lichtschnitt- und Phasenshift-Sensoren. Die Simulationen laufen vollständig auf der GPU mit enorm hohen Geschwindigkeiten. In diesem Beitrag wird die Funktionsweise der Verfahren erläutert und es werden Zeitabschätzungen für die Verfahren gegeben. Anhand zweier Beispiele wird schließlich die Anwendung der Simulationen zur Ansichtenplanung bei optischen Prüfaufgaben vorgestellt.

Keywords: Simulation, Ansichtenplanung, GPU, Phasenshift, Laserlichtschnitt

Motivation und Problemstellung

Mit Einsetzen der vierten industriellen Revolution befindet sich die Produktion von Waren in einem starken Wandel. Eine derzeit absehbare Tendenz ist, dass sich die Variantenvielfalt der Produkte kontinuierlich steigern wird, wobei Losgrößen bei der Herstellung durch eine zunehmende Individualisierung stark sinken [1]. Dies hat direkten Einfluss auf die Planung und den Betrieb optischer Inspektionssysteme. Einerseits sinken damit die zur Verfügung stehenden Ressourcen, um Inspektionen vorzubereiten und zu parametrisieren, andererseits stellt der Variantenreichtum eine enorme Hürde dar, um optische Prüfungen wirtschaftlich betreiben zu können.

Wir schlagen daher vor, in diesen Fällen modellbasierte Prüfungen durchzuführen. Die Grundidee besteht darin, anhand von CAD-Modelldaten der zu prüfenden Objekte und mathematischen Modellen der eingesetzten Sensoren Inspektionen bereits offline zu planen und während des Betriebs synthetische Messdaten als Vergleichsdaten zu erzeugen. Dazu spielt eine geeignete Simulation von Messdaten eine entscheidende Rolle, weshalb wir hier ein Framework vorstellen, mit dem dies für optische Sensoren realitätsnah durchgeführt werden kann.

Stand von Literatur und Technik

Die Simulation von optischen Sensoren ist ein sehr ausgiebig erforschtes Gebiet der Computergrafik. Für Kameras existieren eine

Vielzahl von Ansätzen, mit denen künstliche Szenen generiert werden. Hierbei muss abgewogen werden, ob eine Szene möglichst realitätsnah oder möglichst schnell dargestellt werden soll. Für realitätsnahe Darstellungen sind physikalisch korrekte Simulationsumgebungen erforderlich wie beispielsweise in [2] beschrieben. Andere, zum Teil GPU-gestützte Verfahren arbeiten deutlich schneller, erzeugen naturgemäß nur vereinfachte Darstellungen [3].

Im Bereich der 2,5-dimensional messenden Sensoren existieren für das Laserlichtschnitt-Verfahren bereits eine Vielzahl an Lösungsansätzen: Sie teilen sich im Wesentlichen in Verfahren auf, die die erwarteten Oberflächenpunkte explizit berechnen [4] oder durch Raytracing implizit simulieren [5]. Für Sensoren, die flächig 3D-Daten durch strukturierte Beleuchtung rekonstruieren, sind Simulationsverfahren bekannt, die im ersten Schritt virtuelle Kamerabilder erzeugen und die eigentliche 3D-Rekonstruktion dann auf diesen ausführen [6].

Modellierung von schiefen Sichtvolumina

Um das Verhalten optischer Systeme simulieren zu können, muss deren Abbildungsverhalten hinreichend genau approximiert werden. Das weit verbreitete Lochkameramodell [7] hat sich hier als praktisch erwiesen, deckt jedoch nicht alle Kamera-Objektiv-Anordnungen ab. Beispielsweise lässt sich damit die Scheimpfluganordnung, wie sie in großem

Umfang in Laser-Lichtschnittsensoren eingesetzt wird, nicht abbilden [8].

In der Computergrafik hingegen hat sich das Einheitssichtvolumen (Normalized Device Coordinates) mit den Eckpunkten mit den Eckkoordinaten $[-1, -1, -1]$ bis $[1, 1, 1]$ durchgesetzt [9]. Wir suchen daher nach einem universellen Lösungsweg, ein Sichtvolumen, das aus dem Datenblatt des Sensorherstellers entnommen oder durch Kamerakalibrierung berechnet wird, durch Projektion auf das Einheitssichtvolumen zu transformieren.

Gegeben seien zwei Punktemengen als Matrizen mit den homogenen Punktekoordinaten in den Spalten

$$X = \begin{bmatrix} x_1^1 & \dots & x_1^k \\ \vdots & \ddots & \vdots \\ x_n^1 & \dots & x_n^k \\ 1 & \dots & 1 \end{bmatrix} \text{ und } Y = \begin{bmatrix} y_1^1 & \dots & y_1^k \\ \vdots & \ddots & \vdots \\ y_m^1 & \dots & y_m^k \\ 1 & \dots & 1 \end{bmatrix},$$

und gesucht ist die $(m+1) \times (n+1)$ Matrix der homogene Abbildung A , die X auf Y bis auf Skalierung abbildet

$$AX = \begin{bmatrix} a_1^1 & \dots & a_1^{n+1} \\ \vdots & \ddots & \vdots \\ a_{m+1}^1 & \dots & a_{m+1}^{n+1} \end{bmatrix} X = Y \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_k \end{bmatrix}$$

In dieser Gleichung lassen sich beide Seiten transponieren und spaltenweise aufschreiben

$$X^T \begin{bmatrix} a_i^1 \\ \vdots \\ a_i^{n+1} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_k \end{bmatrix} \begin{bmatrix} y_i^1 \\ \vdots \\ y_i^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} y_i^1 \\ \vdots \\ y_i^k \end{bmatrix}$$

mit $i = 1, \dots, m+1$. Vertauschen der Rolle von s_2, \dots, s_k und y_i^1, \dots, y_i^k ergibt

$$X^T \begin{bmatrix} a_i^1 \\ \vdots \\ a_i^{n+1} \end{bmatrix} - \begin{bmatrix} 0 & \dots & 0 \\ y_i^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & y_i^k \end{bmatrix} \begin{bmatrix} s_2 \\ \vdots \\ s_k \end{bmatrix} = \begin{bmatrix} y_i^1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Daraus kann ein Gleichungssystem für die Unbekannten s_2, \dots, s_k und a_i^1, \dots, a_i^k zusammengestellt werden

$$\begin{bmatrix} 0 & \dots & 0 \\ y_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & y_1^k \\ \vdots & & \vdots \\ 0 & \dots & 0 \\ 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} X^T \dots 0 \begin{bmatrix} s_2 \\ \vdots \\ s_k \\ a_1^1 \\ \vdots \\ a_1^{n+1} \\ \vdots \\ a_{m+1}^1 \\ \vdots \\ a_{m+1}^{n+1} \end{bmatrix} = \begin{bmatrix} y_1^1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

welches mit einem beliebigen Verfahren gelöst werden kann. In einigen unserer Beispiele war es sinnvoll die erste gefundene Lösung iterativ zu verbessern, siehe z.B. Kapitel 2.5 in [10].

Für die Modellierung der Sichtpyramide können in einem allgemeinen Fall Eckpunkte des Bildfeldes im Raum und das Projektionszentrum gegeben sein. Um diese mit Hilfe von Koordinaten zu beschreiben, wird im Projektionszentrum ein Koordinatensystem festgelegt, z.B. wie in Abb. 1 mit Y nach oben und Z von der Sichtpyramide weg.

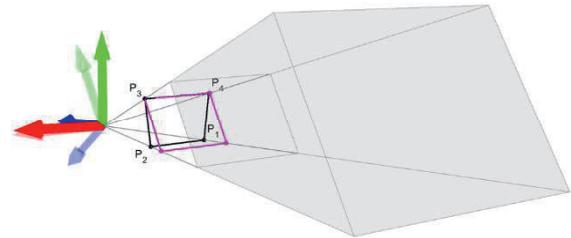


Abb. 1: Konstruktion der Sichtpyramide aus vier gegebenen Eckpunkten (schwarz) über ein Hilfskoordinatensystem (transparent) mit rechteckigem Sichtfeld (magenta).

Nun können die Eckpunkte, die z.B. auf einer Projektionsebene gegeben sind, auf die Ebene $z = -1$ projiziert werden. In der Skizze sind diese Punkte mit P_1, \dots, P_4 bezeichnet. Sie sind folgendermaßen sortiert links unten, rechts unten, rechts oben und links oben. Im allgemeine Fall bilden sie kein Rechteck. Es muss folglich noch die Abbildung bestimmt werden, die diese Punkte auf ein Rechteck abbildet. Dazu verwenden wir obige Methode und ermitteln die 3×3 Matrix A für

$$X = \begin{bmatrix} p_1^1 & \dots & p_1^4 \\ p_2^1 & \dots & p_2^4 \\ -1 & \dots & -1 \end{bmatrix} \text{ und } Y = \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}.$$

Diese Matrix beschreibt eine Drehung und eine Projektion $A = CD$, wobei die Projektion eine obere Dreiecksmatrix ist. Um diese Zerlegung mit verfügbaren Methoden zu gewinnen, verwenden wir die Householder QR-Zerlegung. Da die Reihenfolge Drehmatrix und obere Dreiecksmatrix nicht die gewünschte ist nutzen wir eine Hilfsmatrix

$$E = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix},$$

und zerlegen $M^T E^T = QR$. Daraus folgt $M = (E^T R^T E)(E^T Q^T)$ und somit $C = E^T R^T E$ und $D =$

$E^T Q^T$. Die Projektionsmatrix wird mit $C = C/c_3^3$ normalisiert. Eventuelle negative Elemente auf der Diagonale werden durch folgende Prozedur korrigiert. Falls $c_1^1 < 0$ gilt, werden die Matrizen folgendermaßen korrigiert $C = C \text{diag}([-1, -1, 1])$ und $D = \text{diag}([-1, -1, 1]) D$. Falls $c_2^2 < 0$ gilt, werden $C = C \text{diag}([1, -1, -1])$ und $D = \text{diag}([1, -1, -1]) D$ gesetzt. Damit hat C die typische Form einer Kameramatrix.

Nun können wir die Eckpunkte in das Koordinatensystem drehen, in welchem die Kameramatrix angewendet wird. In Abb. 1 ist dieses Koordinatensystem transparent dargestellt. In diesem Koordinatensystem können die Eckpunkte $X' = DX$ ebenfalls auf die Ebene $z = -1$ projiziert werden, die wir mit X'' bezeichnen und mit $X''' = D^T X''$ in das ursprüngliche Koordinatensystem zurück drehen. In der Skizze sind diese Punkte in magenta dargestellt und es ist erkennbar, dass sie ein Rechteck bilden. X''' kann nun mit den gewünschten Entfernungen skaliert werden, um die nahe und ferne Begrenzung der Sichtpyramide zu konstruieren.

Die Abbildung auf das Einheitssichtvolumen wird nun nach obiger Methode mit den Matrizen

$$\begin{bmatrix} s_{\text{nah}} X''' & s_{\text{fern}} X''' \\ 1 & \dots & 1 & 1 & \dots & 1 \end{bmatrix}$$

und

$$\begin{bmatrix} -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

gefunden.

Framework zur Simulation von Messdaten

Um die Frage zu beantworten, wo ein optischer Sensor platziert werden muss, um ein Merkmal ausreichend gut erfassen zu können, kann ein „Generate and Test“-Ansatz [11] verfolgt werden. Dabei wird der unendlich große Lösungsraum der Sensorposition diskretisiert und für jede Position eine Messdatensimulation durchgeführt. Deren Ergebnis wird dann anhand einer Qualitätsfunktion bewertet. Dabei gilt im Allgemeinen, dass mehr potenzielle Sensorpositionen bewertet werden können, je schneller die Simulation durchgeführt werden kann.

Im Bereich der 2D-Bildverarbeitung werden in vielen Anwendungen die Kanten eines Objekts zur Lokalisierung und Bewertung herangezogen. Das hier vorgestellte Framework ist in der Lage, für Kameras, deren Abbildungseigenschaften nach dem obigen Verfahren definiert werden können, mögliche Objektkanten auf der GPU zu erzeugen. Dazu wird das in [12] vorgestellte Verfahren

aufgegriffen, bei dem durch ein 2-Pass-Rendering synthetische Objektkanten erzeugt werden. Diesem Verfahren liegen folgende Annahmen zugrunde. Erstens: Flächen, die in unterschiedlichen Entfernungen zur Kamera liegen, erzeugen an dem Tiefensprung eine Kante. Zweitens: Flächen, die Unstetigkeiten in der Oberflächennormale aufweisen, erzeugen an diesen sich abbildende Kanten. Und drittens: Unterschiedliche Objekte, die aneinandergrenzen, erzeugen möglicherweise Kanten.

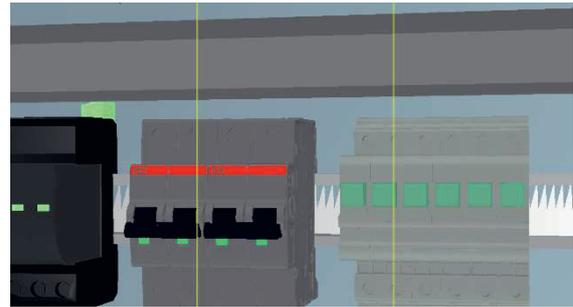


Abb. 2: CAD-Modell einer Baugruppe.



Abb. 3: Zwischenschritte zur Feature-Extraktion des Modells aus Abb. 2: links Tiefenpuffer, mittig Normalenpuffer, rechts Objekt-ID-Puffer.

Zu Umsetzung wird ein zweistufiges Render-Verfahren genutzt. Mit den erforderlichen Kameraparametern wird eine Szene im ersten Pass so gerendert, dass die Entfernung zur Kamera, die Oberflächennormalen und die Objekt-ID in drei Puffer geschrieben werden (Multi-Target-Rendering, s. Abb. 3). In einem zweiten Render-Pass wird jeder Puffer pixelweise mit einem Differenzoperator gefaltet und eine Schwellwertbildung durchgeführt (s. Abb. 4). Die Vereinigung dieser drei Kantenmengen ergibt dann das simulierte Ergebnisbild (Abb. 5).

Im Bereich der 3-dimensionalen Objekterfassung spielt die aktive Triangulation insbesondere mit dem Laser-Lichtschnitt- und der Streifenprojektion eine wichtige Rolle. Beim Laser-Lichtschnitt wird das Licht einer Laserdiode über eine geeignete Optik zu einer Laserebene aufgeweitet und auf das Zielobjekt

projiziert. Das dabei diffus reflektierte Licht wird in einer Kamera als Laserlinie abgebildet. Ist die Lage der Laserebene zur Kamera bekannt, lässt sich entlang der abgebildeten Linie über Triangulation die Oberfläche des Objekts diskret rekonstruieren.

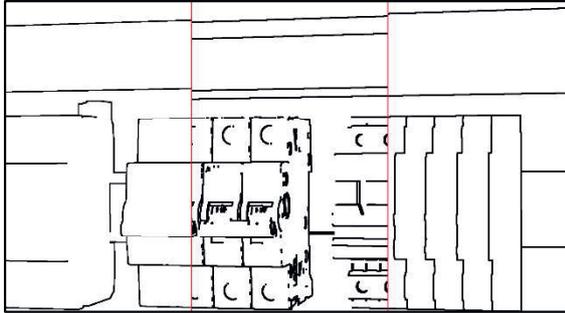


Abb. 4: Extrahierte Kanten aus den Zwischenpuffern.

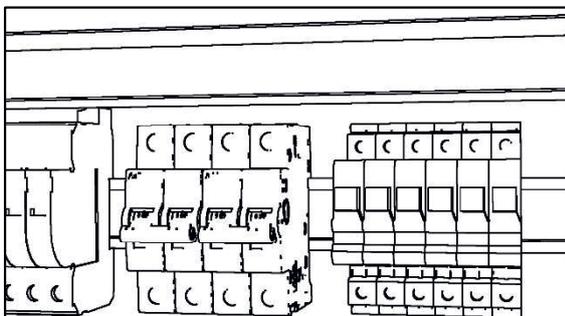


Abb. 5: Vereinigung aller Kanten aus den Zwischenpuffern.

Für eine realitätsnahe Simulation setzt das hier vorgestellte Framework das sogenannte Shadow-Mapping ein [13]. Mit diesem Verfahren werden in der Computergrafik Schatten durch Punkt- und Richtungslichtquellen erzeugt. Zur Simulation des Laserlichtschnitts wird in einem Render-Verfahren mit zwei Pässen entgegen dem eigentlichen Strahlenverlauf die Szene zunächst aus Sicht der Kamera beleuchtet und anschließend aus Sicht des Lasers beobachtet. Im ersten Pass wird die virtuelle Kamera mit den Abbildungseigenschaften der realen Kamera parametrisiert und die Szene aus dieser Perspektive gerendert. Dabei werden lediglich die Tiefenwerte der Szene gespeichert. Für den zweiten Pass führen wir die sogenannte Laser-Kamera ein. Hierbei handelt es sich um eine als Lochkamera modellierte Zeilenkamera, deren horizontaler Öffnungswinkel dem der Laserebene entspricht. Die Laserkamera besitzt eine Auflösung von $1 \times N$, wobei N die typischerweise angegebene Punktauflösung des Sensors ist. Aus Sicht des Lasers wird im zweiten Pass die Szene wiederholt gerendert und das Shadow-Mapping anhand der

Tiefenwerte aus dem ersten Pass durchgeführt. Dadurch werden abgeschattete Bereiche gefiltert und es bleiben die Oberflächenpunkte, die sowohl von der Laserebene geschnitten als auch von der Kamera aus gesehen werden übrig. Die 3D-Koordinaten werden zusammen mit den Oberflächennormalen als Ergebnis ausgegeben (s. Abb. 6). Anhand der Oberflächennormale können anschließend Bewertungsfunktionen beispielsweise die Qualität der zu erwartenden Messpunkte vorherbestimmen.

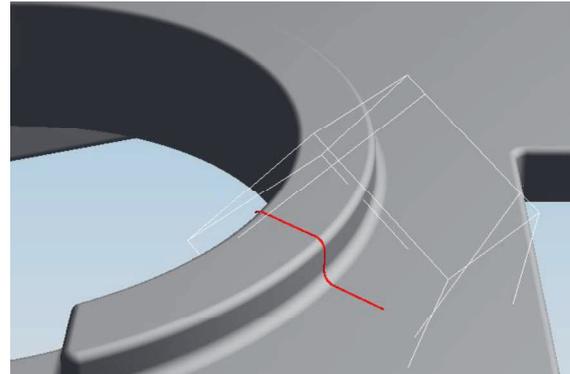


Abb. 6: Simulierter Laserlichtschnitt mit einer Auflösung von 1024 Punkten.

Soll ein Objekt flächig erfasst werden, bietet sich der Einsatz von Sensoren an, die nach dem Prinzip der strukturierten Beleuchtung arbeiten (Streifenlicht- bzw. Phasenshift). Hierbei wird statt einer Linie eine Serie von Bildern auf die Objektoberfläche projiziert. Die Bilderserie ermöglicht eine eindeutige Codierung mehrerer Lichtebenen und eine entsprechend großflächige 3D-Rekonstruktion.

In der Praxis werden dabei zwei unterschiedliche Ansätze verfolgt. Im ersten Fall wird das projizierte Muster genutzt, um der Objektoberfläche zusätzliche und robuste Merkmale hinzuzufügen. Es erfolgt dann eine 3D-Rekonstruktion über ein Kamerapaar durch eine Stereo-Korrespondenzsuche. Im zweiten Fall wird der Strahlengang des Projektors kalibriert. Dadurch kann eine Triangulation direkt zwischen Projektor und Kamera erfolgen. Hier finden sich in der Praxis Sensoren, die zusätzlich eine zweite Kamera – zum Teil mit unterschiedlichem Basisabstand – besitzen, um die Messdatenaufnahme genauer und robuster zu gestalten.

Die Simulation von Phasenshift-Sensoren ist in dem hier vorgestellten Framework ebenfalls auf der GPU umgesetzt worden. Die Simulation erfordert vier Render-Pässe und erzeugt analog zur Laser-Lichtschnitt-Simulation eine 3D-Punktwolke der zu erwartenden Messdaten. Für die Simulation werden das Sichtvolumen der linken und rechten Kamera sowie des

Projektors benötigt (s. Abb. 7). In den ersten beiden Render-Pässen wird die Szene aus der linken bzw. rechten Kamera gerendert und die Tiefenwerte gespeichert. Damit sind die Oberflächenpunkte bekannt, die theoretisch rekonstruiert werden können. Im dritten Pass wird die Szene aus Sicht des Projektors gerendert und die Tiefenwerte ebenfalls gespeichert. Damit sind die Oberflächenpunkte ermittelt, die vom Projektor bestrahlt werden.

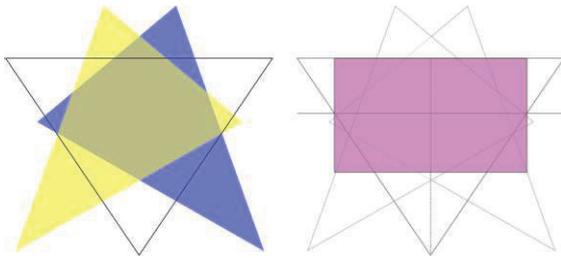


Abb. 7: Sichtvolumen während der Phasenshift-Simulation: links: Kamera und perspektivischer Projektor, rechts zusätzlich der orthogonale Projektor.

Im vierten Schritt findet die eigentliche 3D-Rekonstruktion statt. Hier wird die Szene orthogonal mit der physischen Auflösung des Phasenshift-Sensors aus Sicht des Projektors gerendert. Dies entspricht der 3D-Rekonstruktion der strukturierten Beleuchtung, die im Allgemeinen parallel zur optischen Achse des Projektionssystems durchgeführt wird. Während der Rekonstruktion wird das Shadow-Mapping durchgeführt, indem die Tiefenwerte der Kameraabbildungen aus den ersten beiden Render-Pässen mit den der perspektivischen Projektorabbildung aus dem dritten Pass verglichen wird. Für eine aktive Stereo-Rekonstruktion muss ein Punktkandidat in beiden Kameras sichtbar sein, für eine Projektor-Kamera-Rekonstruktion ist die Sichtbarkeit in einer Kamera bereits ausreichend

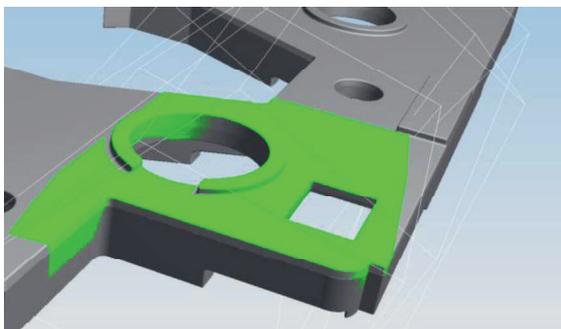


Abb. 8: Ergebnis der Phasenshift-Simulation auf einem Testobjekt.

Für nicht abgeschattete Punkte werden ebenfalls die 3D-Oberflächenkoordinaten und –normalen ausgegeben (s. Abb. 8).

Die Leistungsfähigkeit der Simulationen wurde mit synthetischen Tests ermittelt. Dazu wurden CAD-Modelle von Kugeln mit steigender Anzahl an Dreiecken simuliert und die benötigte Simulationszeit gemessen. Die Tests wurden auf einem Intel Core i7-8700K mit einer Consumer-Grafikkarte Nvidia GeForce GTX 1070 Ti durchgeführt. Die gemessenen Zeiten sind in Abb. 9 dargestellt.

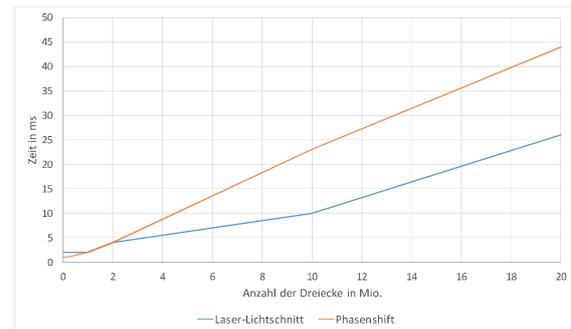


Abb. 9: Benötigte Zeit der 3D-Simulationen in Abhängigkeit von der Anzahl der Dreiecke in einer Szene.

Es zeigt sich, dass für CAD-Modelle mit moderater Komplexität von bis zu 10 Mio. Dreiecken selbst auf Mittelklassegrafikkarten die 3D-Simulationen noch mit Geschwindigkeiten berechnet werden, die eine interaktive Darstellung zulassen.

Das Framework steht unter Windows als C-Bibliothek bereit und kann dank der universellen Definition von Sichtvolumen für eine Vielzahl optischer Sensoren eingesetzt werden.

Anwendung der 2D-Simulation

Eine Inspektionsanlage soll automatisiert die korrekte Montage von Schaltschranktafeln überprüfen. Die Tafeln werden mit einer sehr hohen Variantenvielfalt hergestellt und unterscheiden sich daher zum Teil sehr in ihrer Bestückung. Zur Inspektion soll ein kamerabasiertes Prüfsystem mit einem Roboter über der Schaltschranktafel positioniert und dann eine bildbasierte Auswertung durchgeführt werden. Da die Schaltschranktafeln nur grob zum Roboter positioniert werden können, ist vor der eigentlichen Prüfung eine Referenzierung der Tafel zum Roboter erforderlich. Deren Ergebnis ist die Transformation von Schaltschranktafel- zum Roboterkoordinatensystem. Die Referenzierung soll rein kamerabasiert anhand von natürlichen Merkmalen durchgeführt werden.

Das Ziel besteht nun darin, mit Hilfe einer Messdatensimulation automatisiert geeignete

Sensorpositionen zu bestimmen, aus deren Perspektive der optische Sensor die Merkmale verdeckungsfrei erfassen kann.

Als natürliche Merkmale werden Bohrungen in der Schaltschranktafel, deren 3D-Koordinaten aus dem CAD-System bekannt sind, herangezogen. Als Kamerasensor wird ein Stereo-Kamera-Paar verwendet, dessen intrinsische und extrinsische Parameter zuvor ermittelt wurden. Zur Bestimmung der realen 3D-Bohrungskoordinaten werden diese mit dem Kamera paar aufgenommen, wobei sie sich typischerweise als Ellipsen in den Kamerabildern abbilden. Ist eine Zielbohrung in beiden Kameras sichtbar, kann durch Vorwärtsschnitt der Ellipsenzentren deren 3D-Koordinate bestimmt werden. Nach [14] wird mit den korrespondierenden 3D-Koordinaten aus dem CAD-Modell die Transformation der Schaltschranktafel direkt berechnet. Durch die elektrischen Komponenten und Kabelkanäle kann es jedoch zu Verdeckungssituationen kommen, so dass keine pauschale Positionierung des Stereo-Sensors zu den Bohrungen vorgegeben werden kann. Eine geeignete Aufnahme position muss somit für jede Zielbohrung jeder Schaltschranktafelvariante individuell bestimmt werden.

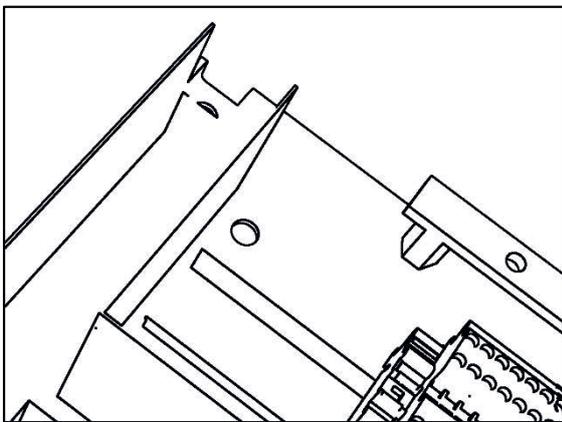


Abb. 10: Simulierte Objektkanten einer Schaltschranktafel. Links ein Kabelkanal, rechts unten Anschlussklemmen, mittig die Zielbohrung.

Zur Lösung dieser Aufgabe wird eine feature-basierte Stereo-Kamerasimulation durchgeführt. Deren Ergebnis sind zwei Abbildungen, in denen die Kanten der gesamten Schaltschranktafel dargestellt werden (s. Abb. 10).

Als prinzipiell geeignet werden Sensorpositionen bewertet, bei denen die Zielbohrung in beiden Kameras vollständig sichtbar ist. Die Qualität einer möglichen Position wird dadurch bestimmt, wie kreisförmig die Bohrung in den Kameras abgebildet wird.

Zur Abtastung des Lösungsraums wird der virtuelle Stereo-Sensor entlang einer Halbkugel oberhalb der Zielkoordinate platziert. Der Halbkugelradius ergibt sich aus dem optimalen Arbeitsabstand des Sensors. Zusätzlich wird bei jeder Position die Rotation des Sensors um seine Sichtachse variiert und die Position des Schaltschranktafelmodells innerhalb des Erwartungsbereichs verschoben. Zur Bewertung werden aus den simulierten Bildern Ellipsen nach [15] extrahiert und – wenn vollständig vorhanden – deren Verhältnis von langer zu kurzer Halbachse als Optimierungskriterium berechnet. Dabei werden Sensorpositionen besser bewertet, in denen die Bohrungsabbildung kreisförmiger ist, um eine möglichst senkrechte Aufnahme position zu erzielen.

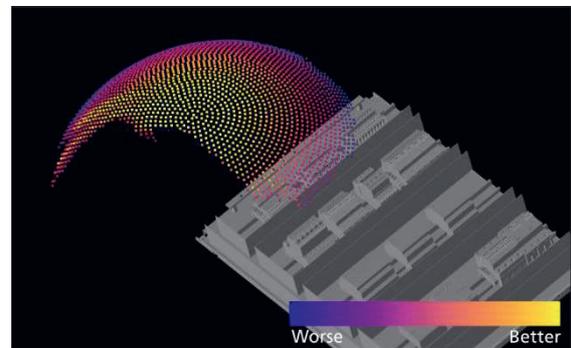


Abb. 11: Visualisierung des abgetasteten Lösungsraums. Farblich codiert die Qualität der zu erwartenden Messdaten aus der jeweiligen Position

Das Ergebnis ist eine Menge an potenziellen Sensortransformationen, bei denen sichergestellt ist, dass eine Zielbohrung aus beiden Kameras möglichst senkrecht und verdeckungsfrei erfasst werden kann.

Anwendung der 3D-Simulation

Ein weiteres Anwendungsbeispiel ist bei einem Sondermaschinenhersteller gegeben. Hier werden Guss- und Schweißteile durch CNC-Maschinen spanend bearbeitet. Charakteristisch ist dabei auch hier, dass die Teile mit einer sehr hohen Variantenvielfalt auftreten. Vor der Bearbeitung sollen die Rohteile einer Qualitätsprüfung unterzogen werden, bei der geometrische Abweichungen identifiziert werden. Die Geometrieprüfung soll für Großbauteile mit Abmessungen bis $5,4 \times 1,7 \times 0,8 \text{ m}^3$ in einer separaten Prü fzelle punktuell stattfinden. Dazu wird ein Streifenlichtsensor eingesetzt, der über ein externes Tracking-System verfügt, mit dessen Hilfe lokale 3D-Sensormessdaten in ein globales

Prüfzellenkoordinatensystem transformiert werden können. Der Sensor ist handgeführt und kommt an kritischen Stellen der zu inspizierenden Teile zum Einsatz. Dort wird das Bauteils flächig erfasst und zur Bestimmung von Geometrieabweichungen eine Abstandsberechnung zum CAD-Modell vorgenommen.

Zur Prüfung wird das Bauteil meist auf einer Palette in der Mitte der Prüfwelle abgesetzt. Als erstes ist deshalb ein Einmessen des Bauteils im Koordinatensystem der Prüfwelle notwendig. Für das Einmessen genügt die 3D-Erfassung weniger Referenzflächen am Bauteil. In diese Messungen kann das 3D-Modell durch Minimierung der Punktabstände zur CAD-Oberfläche eingepasst werden (s. Abb. 12) und damit die Bauteilposition in der Prüfwelle bestimmt werden.

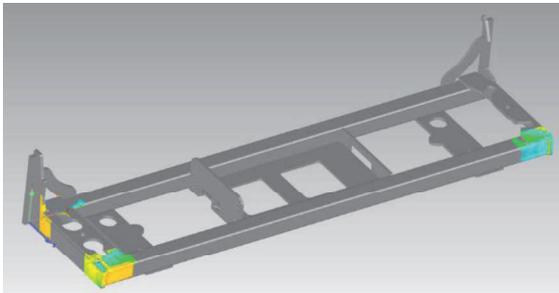


Abb. 12: Registrierung des CAD-Modells in 3D-Messdaten.

Welche Referenzflächen sich für das Einmessen besonders eignen, erfordert ein hohes Maß an Expertenwissen und wird daher von der Arbeitsvorbereitung manuell festgelegt. Im Allgemeinen orientieren sich die Flächen an der Aufspannung in der CNC-Maschine. Zur Markierung der jeweiligen Flächen steht den Mitarbeitern ein virtueller 3D-Pinsel zu Verfügung mit dem sie interaktiv im 3D-Modell Markierungen vornehmen können (s. Abb. 13).

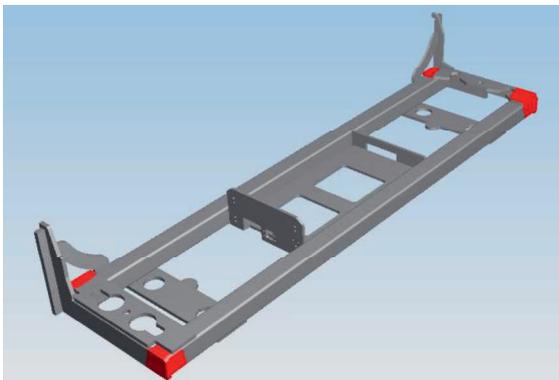


Abb. 13: Mit einem virtuellen 3D-Pinsel markierte Referenzflächen (rot) an einem Bauteil.

Eine geeignete Ausrichtung des Streifenlichtsensors zur Erfassung einer Referenzfläche kann mit Hilfe der zuvor beschriebene Simulationsverfahren automatisch ermittelt werden. Zum einen kann die Arbeitsvorbereitung eventuelle Probleme bei der Erfassung der markierten Referenzfläche erkennen (z.B. schlechte Erreichbarkeit) und zum anderen kann dem Bediener die Sensorausrichtung in einer 3D-Ansicht präsentiert werden, um ihn bei der Durchführung des Einmessens anzuleiten (s. Abb. 14).

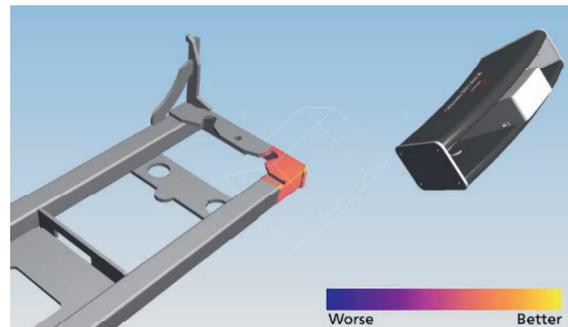


Abb. 14: Visualisierung einer einzunehmenden Sensorposition und der erwarteten 3D-Messdaten. Die Qualität der Messdaten ist farblich codiert.

Zur automatischen Bestimmung der Sensorausrichtung wird der virtuelle Streifenlichtsensor im optimalen Arbeitsabstand von 760mm kugelförmig um das Zentrum der markierten Referenzfläche positioniert. Empirisch wurde ermittelt, dass sich ca. 4000 Positionen auf der Kugel als ausreichend erweisen. Ein weiterer Freiheitsgrad des Sensors ist eine Drehung um die Sichtachse. Der ergonomisch sinnvolle Bereich wird hier durch Drehungen von -30° bis $+30^\circ$ in Schritten von 5° abgedeckt. Zur Bewertung der Sensorpositionen wird für jede Position des diskretisierten Lösungsraums eine 3D-Messdatensimulation durchgeführt und die Qualität der virtuellen 3D-Punkte evaluiert. Dazu werden im ersten Schritt zunächst alle Punkte verworfen, deren Oberflächennormale einen Grenzwinkel zu Kamera und Projektor des Sensors überschreitet. Für die typischerweise auftretenden metallischen Oberflächen in diesem Anwendungsbeispiel hat sich ein Schwellwert von 60° als gut herausgestellt. Zur weiteren Bewertung wird dann die Überlappung der Messdaten mit den markierten Flächen berechnet. Für Sensorpositionen, die eine gleich hohe Abdeckung erreichen, wird zusätzlich die Qualität der zu erwartenden Messdaten bewertet, indem der mittlere Oberflächenwinkel

aller Messpunkte berechnet wird. Dabei gilt, dass eine Sensorposition besser ist, je senkrechter die Flächen zum Sensor stehen. Aufgrund der recht matten Oberflächencharakteristik treten Totalreflexionen sehr selten auf.

Das Ergebnis dieser Simulationen sind Sensorpositionen, die der Bediener während der Bauteilreferenzierung ansteuern soll. Die Positionen werden als 3D-Darstellung visualisiert und dienen als intuitive Anleitung. So ist sichergestellt, dass Messdaten von genau den erforderlichen Referenzflächen aufgenommen werden können, anhand derer die Bauteilposition durch Punkt-zu-CAD-Modell-Registrierung automatisch berechnet werden kann.

Zusammenfassung

In diesem Beitrag wurde ein Framework vorgestellt, mit dem Messdaten von Kameras, Laserlichtschnitt- und Streifenlichtsensoren simuliert werden können. Die Simulation basiert auf den Abbildungs- und Projektionsparametern der Sensoren sowie auf den CAD-Modellen der zu erfassenden Objekte. Für eine möglichst universelle Beschreibung der Kameraabbildung wurde beschrieben, wie ein Mapping von Sichtpyramidenkoordinaten in Einheitskoordinaten stattfinden kann. Die Berechnung der Simulationsdaten zeichnet sich dank vollständiger Implementierung auf der GPU durch eine sehr hohe Geschwindigkeit aus. Dabei ist das Ergebnis der Simulation ausreichend realitätsnah, so dass die Daten beispielsweise für eine Ansichtenplanung verwendet werden können. Es wurden zwei Anwendungen beschrieben, bei denen die Simulation zur Planung von geeigneten Sensorpositionen in 2D und 3D praktisch zum Einsatz kommt.

Danksagung

Diese Arbeit ist im Projekt »3D-KOSYMA« entstanden. Das Projekt wird unter dem Förderkennzeichen 03ZZ0446H im Förderprogramm »Zwanzig20 – Partnerschaft für Innovation« teilweise durch das BMBF gefördert.

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Literaturnachweis

- [1] Buchholz, B. et al.: Digitalisierung industrieller Wertschöpfung. Transformationsansätze für KMU, Berlin, 2017.
- [2] Irgefried, S. et al.: Image formation simulation for computer-aided inspection planning of machine vision systems. In (Beyerer, J.; Puente León, F. Hrsg.): Automated Visual Inspection and Machine Vision II. SPIE, 2017.
- [3] Nilsson, J.; Ericsson, M.; Danielsson, F.: Virtual Machine Vision in Computer Aided Robotics: 2009 IEEE Conference on Emerging Technologies & Factory Automation Piscataway, NJ, 2009.
- [4] Mahmud, M. et al.: 3D part inspection path planning of a laser scanner with control on the uncertainty. In Computer-Aided Design, 2011.
- [5] Malik, A. et al.: A simulation of automatic 3D acquisition and post-processing pipeline. In (Niel, K. S.; Fofi, D. Hrsg.): Image Processing: Machine Vision Applications II. SPIE, 2009.
- [6] Jørgensen, T. B. et al.: Simulation-based Optimization of Camera Placement in the Context of Industrial Pose Estimation. In (Imai, F. H.; Trémeau, A.; Braz, J. Hrsg.): Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP, SciTePress, 2018.
- [7] Xu, G.; Zhang, Z.: Epipolar Geometry in Stereo, Motion and Object Recognition. A Unified Approach. Springer, Dordrecht, 1996.
- [8] Fasogbon, P. et al.: Intrinsic camera calibration equipped with Scheimpflug optical device. In (Meriaudeau, F.; Aubreton, O. Hrsg.): Twelfth International Conference on Quality Control by Artificial Vision 2015. SPIE, 2015.
- [9] Kessenich, J. M.; Shreiner, D.; Sellers, G.: OpenGL programming guide. The official guide to learning OpenGL, version 4.5 with SPIR-V. Addison-Wesley, Boston, 2017.
- [10] Press, W. H.: Numerical recipes in C++. The art of scientific computing. Cambridge Univ. Press, Cambridge, 2005.
- [11] Xi, F.; Shu, C.: CAD-based path planning for 3-D line laser scanning. In Computer-Aided Design, 1999.
- [12] Saito, T.; Takahashi, T.: Comprehensible rendering of 3-D shapes. In (Baskett, F. Hrsg.): SIGGRAPH '90 conference proceedings. ACM Press, New York, 1990.
- [13] Brabec, S.; Annen, T.; Seidel, H.-P.: Practical Shadow Mapping. In Journal of Graphics Tools, 2002.
- [14] Haralick, R. M. et al.: Pose estimation from corresponding point data. In IEEE Transactions on Systems, Man and Cybernetics, 1989.
- [15] Teutsch, C. et al.: Real-time detection of elliptic shapes for automated object recognition and object tracking. In (Meriaudeau, F.; Niel, K. S. Hrsg.): Machine Vision Applications in Industrial Inspection XIV. International Society for Optics and Photonics, 2006.