

Kamerabasierte Absolutwegmessung

Prof. Dr. Stefan May*, Christian Merkl*, Wolfgang Antal* Technische Hochschule Nürnberg, Nürnberg; Dr. Verena Schmidt, Siemens Healthcare GmbH, Kemnath

Im Rahmen einer Konzeptstudie wurde eine günstige Lösung zur Absolutwegmessung mit Hilfe einer Kamera und einem Raspberry Pi realisiert. Die Detektion von Marker und sinusförmigen Grauwertverläufen auf einem Codierband wurde als kombiniertes Bildverarbeitungsmodul implementiert. Die Machbarkeit des Verfahrens und der Robustheit der Algorithmen wurde in Tests nachgewiesen.

Schlagwörter: Absolutwegmessung, Kamera, Codierstreifen, RANSAC-Algorithmus

Innovation to Cost

Innovation to Cost (ItC) nimmt einen wachsenden Stellenwert in der Arbeit einer Innovationsabteilung ein. Der Unterschied zu Design to Cost (DtC) liegt dabei durch intelligente, neuartige Lösungen, Produktkosten zu senken bei gleichbleibender oder sogar steigender Produktqualität.

Neben der technischen Analyse ist es in diesem Fall auch wichtig Kosten neuer Technologien sowie die wirklichen Produkthanforderungen zu prüfen. Aus Mangel an Alternativen werden in der Medizintechnik häufig Sensoren aus der Automatisierungstechnik eingesetzt. Die Langzeitverfügbarkeit dieser Produkte ist gegeben. Zusätzlich bieten sie eine zuverlässige Funktionalität. Gleichzeitig sind die Anforderungen in dieser Branche wesentlich höher als in der Medizintechnik. Dies zeigt sich in der Genauigkeit, der Abtastrate, den Einsatzmöglichkeiten unter rauen Umgebungsbedingungen (z.B. Explosion, Temperatur, Staub) und damit hohen IP-Schutzklassen. Dies führt schließlich zu hohen Produktkosten, die in der Medizintechnik so in vielen Produktbereichen nicht akzeptiert werden können.

Kostendruck bei Röntgengeräten

In mechatronischen Medizinprodukten werden viele Positionen gemessen. Hierfür werden häufig Sensoren wie Absolutwertgeber, die auf Wellen befestigt sind oder Seilzugeber verwendet, deren Preise bei 100€-200€ liegen. Dies ist insbesondere für den Einsatz in Röntgengeräten zu teuer. Aus diesem Grund werden als kostengünstige Alternative auch häufig Potentiometer eingesetzt,

welche inkl. Mechanik ca. 50€ kosten. Der mechanische Verschleiß führt zu Ausfällen, welche jedoch auf Grund des günstigen Preises von Potentiometern im Vergleich zu Absolutwertgebern akzeptiert wird. Es entstehen zusätzliche Kosten durch mechanische Komponenten, welche die zu messenden Bewegungen auf die Drehachse des Sensors übertragen.

Es ist nun notwendig eine zuverlässige Sensorik mit dem Ziel „good enough“ zu entwickeln. So sind in einigen Anwendungen die Anforderungen für die Positionierung geringer als in der Automatisierungstechnik, z.B. liegt die notwendige Genauigkeit im Bereich von mm und Geschwindigkeiten bei ca. 0,1 m/s. Im medizinischen Umfeld ist auch kein Explosionsschutz oder erhöhter Schutz gegen Eindringen von Wasser oder Staub nötig, so dass die resultierenden IP-Schutzklassen wesentlich geringer ausfallen.

Die Absolutwegmessung

Versuchsaufbau

Im Rahmen des Projekts wurde nun eine kamerabasierte Absolutwegmessung basierend auf einer einfachen Web-Cam, einem Raspberry Pi und einem CAN-USB-Adapter implementiert. Der Lösungsansatz ist vergleichbar mit der Nutzung eines Lineals durch einen Menschen. Dabei wird ein Codeband basierend auf einem Datamatrixcode sowie einem sinusförmigen Grauwertkeil durch die Kamera redundant ausgelesen (siehe **Abbildung 1**).

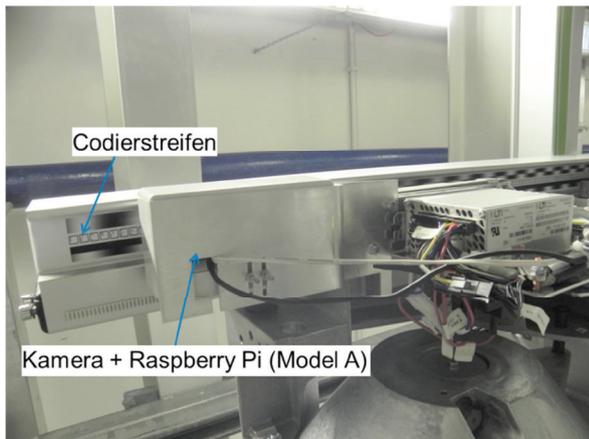


Abbildung 1: Versuchsaufbau mit Kamera und Codierstreifen

Die Kamera wird durch einen umliegenden LED-Ring konstant beleuchtet und ist auf ein Codeband gerichtet. Sie wird relativ zu diesem bewegt. Die Besonderheit gegenüber vorhandenen Lösungen liegt in der Kombination mehrerer Verfahren, um auch bei geringen mechanischen und optischen Anforderungen hohe Genauigkeiten zu erreichen. Das Codeband kombiniert einen Datamatrixcode sowie drei als Graustufen dargestellte Sinuswellen mit einer Wellenlänge, die ungefähr den Bildbereich der Kamera abdeckt.

Abbildung 2 zeigt den Aufbau des Codierstreifens. Für die Auswertung der Sinuswellen werden fünf Bildzeilen herangezogen: Zwei Zeilen liegen am oberen und unteren Bildrand und erfassen einen phasengleichen Sinusverlauf. Zwischen diesen beiden Sinuswellen tritt in Aufnahmen während einer Bewegung eine Phasenverschiebung aufgrund des Rolling-Shutter-Effekts auf. Zwei Zeilen erfassen eine schwarze und eine weiße Referenzlinie, die zur Binarisierung der Bildaufnahme bei unterschiedlichen Lichtverhältnissen und Kameraeinstellungen dient. Der dritte Sinusverlauf dient zur Positionserfassung, da er gegenüber den beiden äußeren Sinuswellen eine Schwebung ausbildet. Diese Phaseninformation ist linear abhängig zur absoluten Position der Vorrichtung.

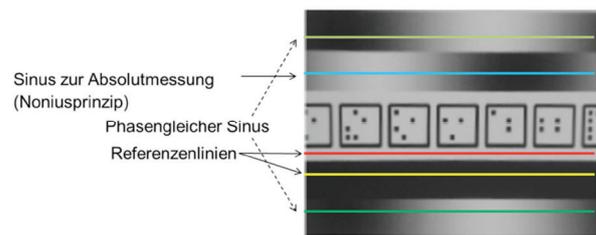


Abbildung 2: Der Codierstreifen

Mit einem 2. Verfahren, dem Datamatrixcode, sind genaue Positionen bestimmbar. Bei höheren Geschwindigkeiten sind aufgrund der Bewegungsunschärfe bei billigen Kameras keine feinen Strukturen mehr auflösbar, so dass der Marker ohne zusätzliche Hardware nur bis ca. 0.2 m/s erfasst werden kann. Die Detektion einer niederfrequenten Sinuswelle ist dann aber immer noch problemlos möglich. Die oben genannte Schwebung ist bei deutlich höheren Geschwindigkeiten noch zu erfassen. Beide Verfahren werden redundant eingesetzt und erhöhen durch ihre Kombination die Robustheit.

Rolling Shutter Effekt

Unter Verwendung des bei einer CMOS-Kamera auftretenden Rolling-Shutter-Effekts wird die Positionsschätzung genauer. Die Zeilen eines Bildes erscheinen bei einer bewegten Aufnahme verschoben (siehe **Abbildung 3**). Dieser Effekt tritt umso deutlicher auf, umso weiter die beiden Zeilen voneinander entfernt sind. Deshalb wurden zwei Zeilen am oberen und unteren Bildrand gewählt.

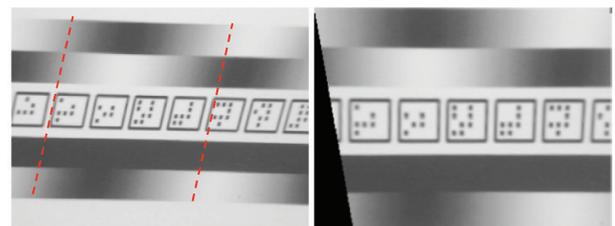


Abbildung 3: Rolling Shutter Effekt

Durch die Rolling-Shutter-Verschiebung tritt eine unterschiedliche Phasenlage (siehe **Abbildung 4**) auf, die linear abhängig von der Geschwindigkeit ist. Bei einer bekannten Latenz zwischen Aufnahme und Signalverarbeitung kann die Geschwindigkeit als Korrekturwert berücksichtigt werden.

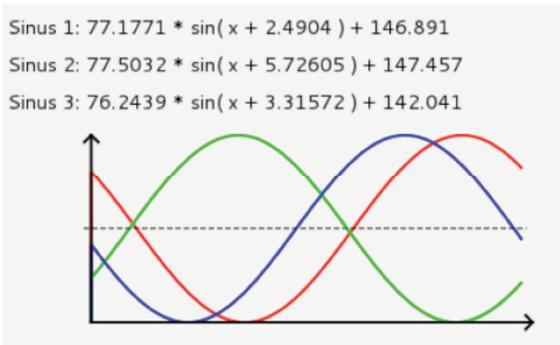


Abbildung 4: Rolling Shutter Effekt im Sinussignal

Robustheit

Die Detektion ist robust gegenüber Verschmutzung, Vergilbung und mechanischen Toleranzen. Dies wird bei der Markerdetektion durch einen Hamming-Abstand von 3 der Marker und einer Auswertung von Sequenzfolgen erreicht. In **Abbildung 5** ist an der ersten Stelle ein falscher Marker aufgeklebt, welcher die Position „2“ repräsentiert. Die anschließenden Marker werden als Positionen „38“ bis „42“ identifiziert. Auf Grund der logischen Lauffolge wird deshalb die erste Position von „2“ auf „37“ korrigiert.

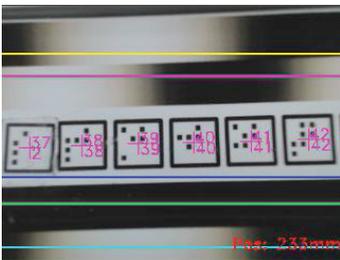


Abbildung 5: „Logisches“ Zählen der Marker

Sinusdetektion

Bei der Wegmessung mit Sinusdetektion wird von einer bekannten Sinus-Frequenz ausgegangen, welche mit Hilfe eines RANSAC-Schätzers und Maximum-Likelihood-Methoden rekonstruiert wird. Dabei werden auch grobe Störungen korrigiert. Marker- und Sinusdetektion werden zudem für eine Plausibilitätsprüfung genutzt.

Dynamische Skalierung

Die aktuelle Lösung erlaubt hohe Ungenauigkeiten bei der Installation, wie z.B. eine Verdrehung der Kamera oder unterschiedliche Abstände zwischen Kamera und Codeband. Aus diesem Grund wurde eine dynamische Skalierung entwickelt (vgl. **Abbildung 6**).

Abbildung 6).

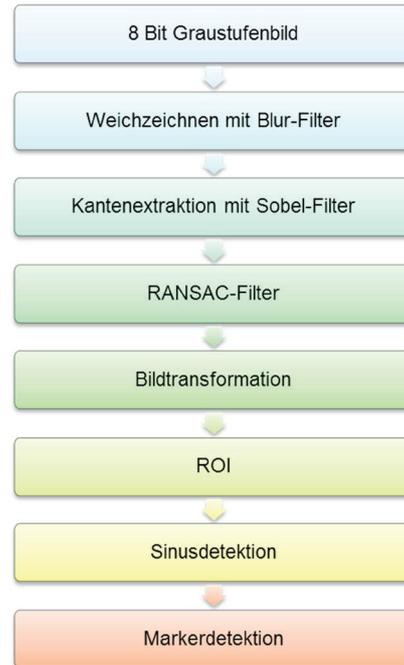


Abbildung 6: Algorithmus zur Absolutwegmessung

Abbildung 7 visualisiert die verschiedenen Auswertestufen anhand eines Beispiels. Zunächst wird ein 8 Bit Graustufenbild eingelesen (a), welches dann mit einem Blur-Filter weichgezeichnet wird (b). Dieser Schritt entfernt hohe Frequenzen und hebt klare Kanten im Bereich der Sinuskurven hervor. Die Kanten werden dann mit einem Sobelfilter gefaltet und anschließend mit dem RANSAC-Schätzer identifiziert. Im letzten Schritt erfolgt die Bildtransformation und somit die Korrektur einer Verkippung. Im Anschluss wird die Region of Interest (ROI) ausgewählt sowie die Sinus- bzw. Markerdetektion durchgeführt.

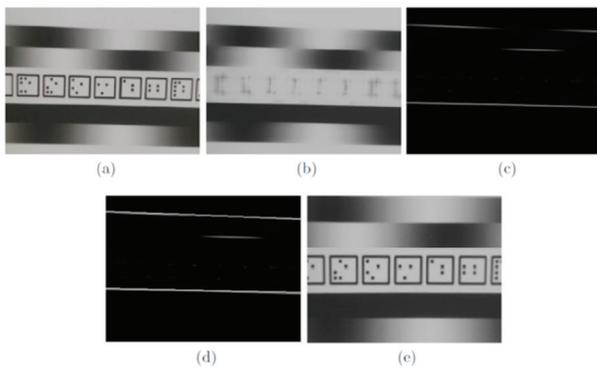


Abbildung 7: Bilder zu den Auswertestufen

Rechenzeitbedarf

Ohne weitere Optimierungen der Algorithmen, wie z.B. Parallelisierung, werden folgende technische Daten erreicht:

- Genauigkeit Marker liegt bei 1 mm.
- Genauigkeit Sinus liegt bei 1 cm.
- Markerdetektion funktioniert bis 0,2 m/s

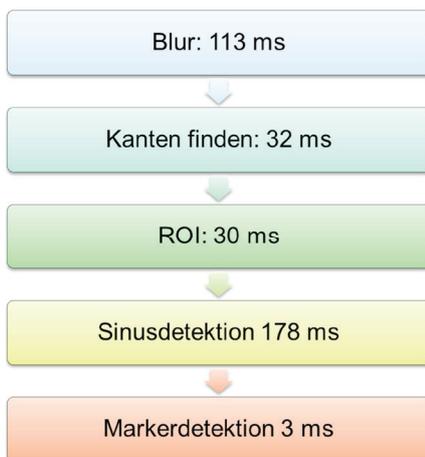


Abbildung 8: zeitliche Bewertung des Algorithmus (Raspberry Pi, Model A)

Eine erste zeitliche Bewertung des Algorithmus ist in **Abbildung 8** zu sehen. Es ist erkennbar, dass die notwendige Rechenzeit für die Sinusdetektion sehr hoch ist. Die Ziel-Abtastrate von 20 Hz wird somit mit diesem System nicht erreicht. Für die weitere Optimierung gibt es folgende Möglichkeiten:

- Erhöhte Anforderungen an die Installation, so dass ein Drehen und Verkippen ausgeschlossen wird – der Blur-Filter und die Kantendetektion würden dann nicht mehr benötigt
- Bei einer ausschließlichen Verwendung der Marker, welche ggf. redundant ausgelegt werden, müsste zusätzlich ein Blitzlicht, welches von der Kamera getriggert wird, angebracht werden. Dies ist nötig, um höhere Geschwindigkeiten zu erzielen. Der Vorteil dieser Lösung liegt in einer reduzierten Rechenzeit.
- Bei Kombination von Marker- und Sinusdetektion ist eine entsprechende Parallelisierung des Algorithmus und der Einsatz leistungsstärkerer Hardware nötig. Der Raspberry Pi 2 bietet zum annähernd gleichen Preis eine deutlich höhere Rechenleistung mit vier CPU-Kernen.

Stoßkanten

Zur Realisierung langer Wegstrecken (bis zu 16m) ist es möglich, dass mehrere Codierstreifen verwendet werden. Bei der Installation kann es dann zu Stoßkanten / Abständen zwischen zwei Codierstreifen kommen. Mit Hilfe eines Werkzeugs werden diese Abstände bei einer Kalibrierfahrt vermessen und hinterlegt. Hierfür wird der Abstand zwischen zwei Markern berechnet.



Abbildung 9: Stoßkantenkalibrierung

Die Positionsbestimmung mittels Marker ist bei einer Stoßkante robuster gegenüber dem sinusförmigen Verlauf. Insbesondere entstehen bei größeren Abständen zwischen zwei Codierbändern Fehler bei der Sinusdetektion.

Vergleich Sinus- zu Markerdetektion

Die Experimente zeigen, dass die beiden implementierten Algorithmen zur Absolutwegmessung Stärken und Schwächen haben. Unter Berücksichtigung der implementierten hohen Robustheit ist das Verhalten beider Verfahren in **Tabelle 1** dargestellt.

Tabelle 1: Vergleich Sinus- zu Markerdetektion

| | Marker | Sinus |
|---|-------------------------|--------------|
| Verfahrgeschwindigkeit | < 0,2 m/s | < 0,7 m/s |
| Genauigkeit | 1 mm | 10mm |
| Stoßkanten | Kein Problem | |
| Kontrasteinflüsse wie Farbe, Vergilbung | Robuster als Sinus | |
| Rechenzeit | 3 ms | 178 ms |
| Notwendige Hardware bei Optimierung | Getriggertes Blitzlicht | CPU-Leistung |

Für eine Produktimplementierung gilt nun abzuwägen, ob eine Kombination beider Algorithmen oder nur die Markerdetektion verwendet werden soll. Der Vorteil der Nutzung beider Varianten liegt im Bereich der Sicherheitsbetrachtung, da hier die Realisierung eines redundanten Systems basierend auf unterschiedlichen Implementierungen möglich ist.

Vorteile der Kameralösung

Im Rahmen der Studie wurde der Nachweis erbracht, dass eine kostengünstige Lösung für die Absolutwegmessung mit Hilfe einer Kamera und eines Codierbandes möglich ist.

Die Vorteile einer kamerabasierten Lösung liegen im Verschleiß. Zunächst ist offensichtlich, dass kein mechanischer Verschleiß am Sensorelement im Vergleich zu einem resistiven Potentiometer auftreten kann. Dies betrifft aber auch den

restlichen mechanischen Aufbau. Es gibt bereits magnetische Potentiometer-Alternativen, die jedoch noch eine Anbindung an das System z.B. über Antriebsriemen oder Zahnräder benötigen, welche wieder einem mechanischen Verschleiß unterliegen. Die Kameralösung dagegen kann ohne mechanische Verschleißteile einfach installiert werden.

Dieses Beispiel zeigt, dass es möglich ist, durch Einsatz vorhandener Hardware, kostengünstige und dennoch zuverlässige Sensorsysteme aufzubauen, die gleichzeitig den mechanischen Verschleiß verringern.

Das Potential von Kameras wird zudem als besonders hoch eingeschätzt. Es ist möglich die vorhandene Hardware durch Veränderung des Algorithmus auf einfachste Art und Weise für weitere Messaufgaben umzubauen. Dies führt zu weiteren Stückzahleneffekten und reduziert gleichzeitig die Komponentenvielfalt.