

Nanopositionier- und Nanomessmaschinen – Herausforderungen bei der Visualisierung großer Messvolumina mit Übersichtsbildern im Gigapixel-Bereich

Nanopositioning and Nanomeasuring Machines - Challenges in Visualizing Large Measurement Volumes with Overview Images in the Gigapixel Range

Oliver Birli, Folker Schwesinger, Prof. Eberhard Manske, Fertigungs- und Präzisionsmesstechnik, Technische Universität Ilmenau, Gustav-Kirchhoff-Straße 1, 98693 Ilmenau, Deutschland, oliver.birli@tu-ilmenau.de

Kurzfassung

Die Orientierung im Messvolumen von Nanopositionier- und Nanomessmaschinen mit großen Messbereichen (z.B. NPMM200 mit 200x200x25 mm Messbereich) ist speziell bei unbekannten Proben anspruchsvoll. Ein grobes Scannen mit AFM- oder Laserfokus-Sensoren (z.B. mit einem Pixelabstand von ca. 500 nm) ist extrem zeitaufwendig. Verwendet man zum Erstellen eines 3D-Übersichtsbildes Mikroskope und Kameras, kann in überschaubarer Zeit ein 3D-Abbild des Messobjektes erstellt werden. Durch eine Konvertierung in spezielle Datenformate ist eine nahezu verzögerungsfreie Navigation innerhalb der Grenzen des Messobjektes möglich. Die Herausforderungen und Probleme bei der Entwicklung derartiger Software bzgl. Rechnerarchitekturen, Datenspeicherung und Parallelisierung werden in dieser Abhandlung näher untersucht.

Abstract

Orientation in the measuring volume of nanopositioning and nanomeasuring machines with large measuring ranges (e.g. NPMM200 with 200x200x25 mm measuring range) is challenging especially for unknown samples. A coarse scanning with AFM or laser focus sensors (e.g. with a pixel pitch of about 500 nm) is extremely time consuming. If microscopes and cameras are used to create a 3D overview image, a 3D image of the measured object can be created in a manageable amount of time. Through a conversion into special data formats, an almost instantaneous navigation within the boundaries of the measuring object is possible. The challenges and problems in the development of such software regarding computer architectures, data storage and parallelization are examined in more detail in this paper.

1 Einleitung

Für die an der Technischen Universität entwickelte Nanopositionier- und Nanomessmaschine NPMM200 mit einem Messbereich von 200x200x25 mm war es notwendig, die zur Steuerung der Anlage, Durchführung von Messungen und Visualisierung von Messergebnissen notwendige Software selbst zu entwickeln. Kommerzielle Software war nicht verfügbar, da der Leistungsumfang der NPMM200 weit über den Stand der Technik hinausging und eine Entwicklung von Softwarekomponenten notwendig wurde, die stark an die spezielle Hardwarearchitektur der Anlage angepasst war. Neben einer echtzeitfähigen FPGA-basierenden Firmware (LabVIEW), Server-Client basierten Programmen für Umweltdaten und zur Steuerung integrierter Hardwarekomponenten, wurden auch Python-APIs entwickelt, mit denen der Anwender die Anlage mit Hilfe beliebiger Python-IDEs bedienen kann.

Dieser Artikel beschäftigt sich mit Herausforderungen bei der Optimierung der Software zur Verarbeitung und Visualisierung der Messergebnisse bezüglich unterschiedlicher Rechnerarchitekturen, der Datenspeicherung und Parallelisierung der seit 2012 entwickelten Windows-Software.

2 Aufbau und Stand der Technik

Der Aufbau besteht aus der an der Technischen Universität Ilmenau entwickelten Nanopositionier- und Nanomessmaschine NPMM200, einem NIKON Kompaktmikroskop mit 10-fach SLWD Objektiv und einer 6 Megapixel Allied Vision GigE-Kamera mit einem 1" Sony ICX694 Sensor (**Bild 1**). Die Steuerung der NPMM200 und des Kamerasystems erfolgen über Python-APIs. Eine Software für das Stitching der aufgenommenen Bilder ist nicht notwendig, die hohe Genauigkeit der NPMM200 erlaubt das spätere, nahtlose Zusammenfügen der Bilder.

In dieser Hardwarekonfiguration entstehen beim Scannen des gesamten Messbereichs ca. 55.000 Bilder mit einer Größe von ca. 4 Megapixel (zusammen 220 GB). Ein Zusammenfügen bzw. Stitching der Bilder mit kommerzieller Software war nicht möglich, bereits nach wenigen 100 Bildern konnten keine weiteren Dateien importiert werden und der Vorgang wurde abgebrochen. Die Umwandlung der Bilder bzw. Bildinformationen in numerische 3D-Koordinaten hat das Datenvolumen um ein Vielfaches vergrößert. Die aufgenommenen ca. 200 Mrd. Pixel ergeben in einer binären Darstellung ca. 2,4 TB (als FLOAT-Werte)

und ca. 4,8 TB (als DOUBLE-WERTE) große Messwertdateien. Abhängig vom Messbereich kann mit FLOAT-Werten (max. Auflösung 7 digits) der vollständige Messbereich der NPM200 nicht abgebildet werden.

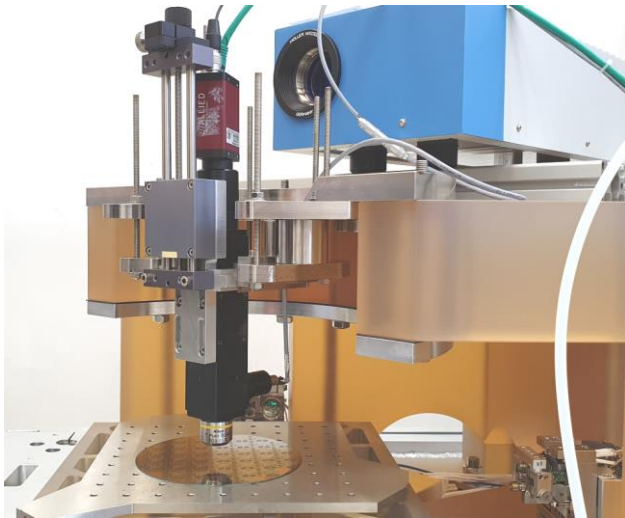


Bild 1 NPM200 mit Sensorsystem (Mikroskop und Kamera) und Z-Verstellung

3 Bildaufnahme

Für ein besseres Verständnis der späteren Messdatenverarbeitung werden in diesem Abschnitt die einzelnen Schritte der Bildaufnahme dargestellt. Das Mikroskopsystem (einschließlich Kamera) besteht aus den oben aufgeführten Komponenten. Die Gigabit-Schnittstelle der Kamera begrenzt die Anzahl der Bilder auf 16 pro Sekunde. Relevant ist diese Begrenzung bei der Fokussuche oder wenn das System mit einem Mirau-Objektiv zur Weißlichtinterferometrie verwendet wird. Die Beleuchtung wird über eine LED-Lichtquelle realisiert. Die Bildaufnahme unterteilt sich in drei Arbeitsschritte:

1. Ermittlung des Abbildungsmaßstabes und der Verdrehung zwischen Arbeitsplattform und Kamerasystem
2. Durchführen einer Fokussuche und Erzeugung von Stützstellen zum Ausgleich von Verkippungen zwischen Arbeitsplattform/Messobjekt und Kamera
3. Automatische Aufnahme der Bilder (mittels Python-Skripten) über den gesamten Messbereich

Für die Ermittlung des Abbildungsmaßstabes und der Verdrehung zwischen Arbeitsplattform und Kamerasystem kommen eigene, mittels Laserschreiben hergestellte Kalibriermarken auf Glassubstraten zum Einsatz [1].

Die Kalibriermarke wird manuell vorpositioniert und der Fokus grob eingestellt. Ein Python-Skript steuert den weiteren Ablauf. Über eine automatische Fokussuche wird der korrekte Schärfentiefebereich ermittelt und auf die Kalibrierstrukturen scharf gestellt. Anhand der Linienstrukturen können die kreisrunden Kalibriermarker automatisch in den oberen linken Bildbereich der Kamera verschoben

werden. Durch ein definiertes Verschieben (z.B. 650 µm) der Kalibriermarke innerhalb eines Kamerabildes in x- und y-Richtung können Abbildungsmaßstab und Verdrehung berechnet werden (mittels Python-Skript und OpenCV). Die Ergebnisse in Form der Kameraeigenschaften, Achseneigenschaften für X und Y, sowie die Bildeigenschaften werden in einer JSON-Datei gespeichert und von den Messskripten automatisch importiert (**Bild 2**).

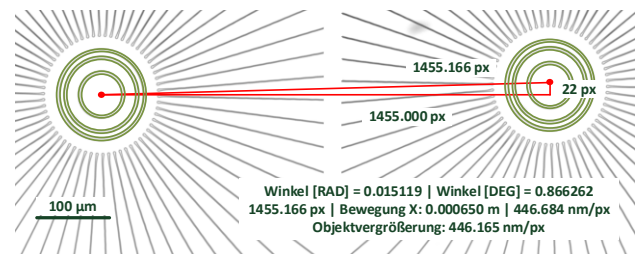


Bild 2 Mit dem Kameramikroskop der NPM200 aufgenommene Bilder der Kalibriermarke (rot gekennzeichnet, die mit OpenCV ermittelten Mittelpunkte der Kreise)

Im zweiten Schritt muss die Verkippung zwischen Arbeitsplattform (mit dem Messobjekt) und der Kamera minimiert werden. Bei einem Messbereich von 200x200 mm kann es vorkommen, dass das Messobjekt unter ungünstigen Befestigungsverhältnissen durchhängt oder schief auf der Arbeitsplattform liegt. Das kann einerseits durch eine Justage oder durch ein kontinuierliches Nachführen der Z-Achse bei der Bildaufnahme beseitigt werden. Die 12 µm große Schärfentiefe des Kamerasystems erlaubt also nicht die Aufnahme aller Bilder in einer fest definierten Z-Ebene.

Dafür wurde das Messskript um die Funktionalität einer Fokussuche an definierten Positionen innerhalb des Messbereichs erweitert. Im einfachsten Fall wird an fünf Positionen (oben links, oben rechts, unten links, unten rechts und in der Mitte) eine Fokussuche durchgeführt und die Z-Position des schärfsten Bildes innerhalb des Bilderstapels abgespeichert. Für die Ermittlung der Schärfe eines Bildes werden Funktionen von OpenCV verwendet. Für jede XY-Position wird ein Grobscan (±150 µm Z-Bewegung) und ein Feinscan (±50 µm Z-Bewegung) durchgeführt. Der Feinscan startet an der Z-Position des schärfsten Bildes vom Grobscan.

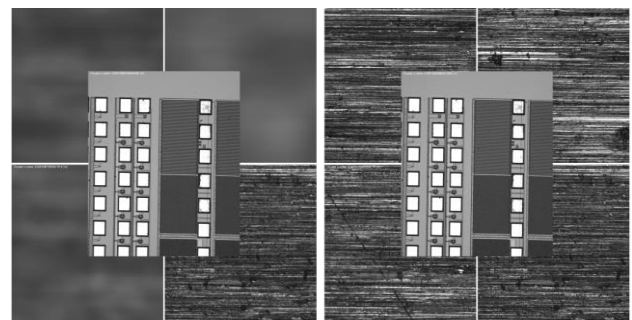


Bild 3 Aufnahmen vom gesamten Messbereich an fünf Positionen. Auf fester Z-Ebene (links) und nach automatischer Fokussuche (rechts)

Die Z-Koordinaten an anderen XY-Positionen werden beim Messen entsprechend interpoliert. Dieses Verfahren ist schnell, aber relativ ungenau, es entstehen u.U. viele unscharfe Aufnahmen, abhängig vom Messobjekt und dessen Justage auf der Arbeitsplattform (**Bild 3**).

Bessere Ergebnisse erhält man durch Verfeinerung des Gitters mit Stützstellen. Der Anwender kann im Python-Skript die Anzahl der Stützstellen in X- und Y-Richtung definieren, der Messbereich wird aus der Messaufgabe übernommen. Anschließend wird der Bereich der Stützstellen mathematisch über den eigentlichen Messbereich hinaus erweitert, einerseits um die Verdrehung zwischen Kamera und Arbeitsplattform auszugleichen und andererseits um Sprünge bei der Interpolation der XY-Positionen zwischen den Stützstellen zu vermeiden (**Bild 4**).

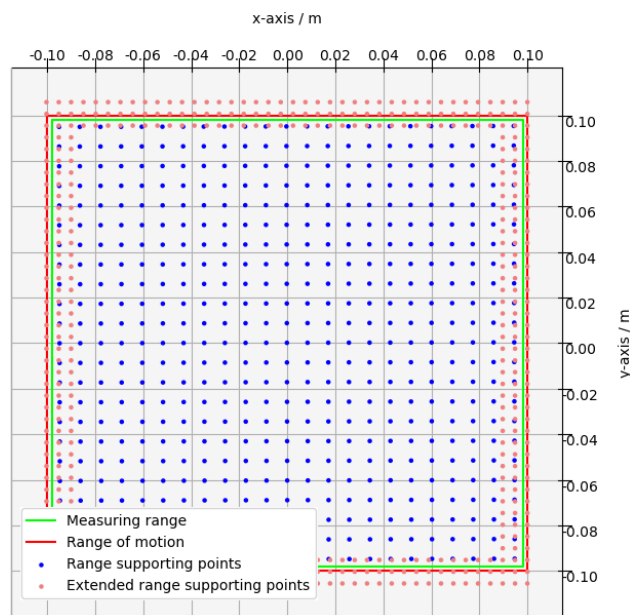


Bild 4 Berechnete Positionen der Stützstellen, an denen eine Fokussuche durchgeführt wird (blaue Punkte)

Zu erkennen ist auch eine leichte Verdrehung zwischen Kamera und Arbeitsplattform. Für die korrekte Ermittlung der Z-Koordinaten an den 576 Stützstellen benötigt die NPMM200 ca. 12 Stunden und nimmt während der 576 Fokussuchen ca. 125.000 Bilder auf.

Im abschließenden Schritt werden alle 55.000 Bilder aufgenommen. Dazu wird jeweils die exakte XYZ-Position berechnet, die NPMM200 fährt die Position an und die Kamera nimmt das Bild auf. Der Vorgang zur Aufnahme eines Bildes dauert ca. 1,2 Sekunden, alle Aufnahmen für den gesamten Messbereich benötigen ca. 19 Stunden.

Durch Reduzierung der Stützstellen kann die Messzeit bei Bedarf signifikant verkürzt werden, bei zu wenig Stützstellen besteht allerdings die Gefahr unscharfer Aufnahmen. In der aktuellen Version der Python-Skripte entstehen parallel zu den Bitmap-Bildern noch einmal die gleiche Anzahl an NUMPY-Arrays, die für die Weiterverarbeitung im TILE-Format-Konverter anstelle der Bitmap-Dateien verwendet werden. Die Bilder dienen nur der visuellen Kontrolle des

Aufnahmeprozesses und können anschließend gelöscht werden. Der gesamte Vorgang der Fokussuche und Bildaufnahme erzeugt ca. 252.000 Dateien in einem Gesamtvolumen von ca. 890 GB.

4 Konvertierung und Visualisierung der Messdaten

Da eine Visualisierung der Messdaten mit kommerzieller Software nicht möglich war, wurde ab 2012 eine eigene Software entwickelt. Sie dient der Visualisierung des kompletten Messvolumens und beinhaltet einfache Funktionen zur Messung von Punktabständen. Gleichzeitig können kleinere Teilbereiche exportiert werden, um diese in kommerzieller Software detaillierter zu untersuchen.

Terabyte große Messwertdateien am Stück zu visualisieren war nicht möglich. Auch die Anordnung von Punkten in den Messwertdateien kann nicht eindeutig definiert werden, diese ist abhängig von Messprogrammen und verwendeten Sensoren. Ein gezieltes Nachladen benötigter Messwerte ist aus diesem Grund nicht möglich, die Messwerte werden ungeordnet in den Messwertdateien abgespeichert.

Aus diesem Grund werden vor der Visualisierung die Messdaten in ein gut darstellbares Format konvertiert. Es wurde ein Dateiformat entwickelt (einschl. Header), in das die Messwertdateien nach dem Empfang der Daten konvertiert werden. Dazu wurde das in [2] und [3] beschriebene Verfahren eines TILE-Servers für 2D Rasterdaten um die dritte-Dimension erweitert und in einem TILE-Format-Konverter implementiert. Die Daten des gesamten Messbereichs werden dazu in Würfeln mit fest definierten Kantenlängen (sogenannten TILES) gespeichert. Auf der untersten Ebene gibt es $n \times m$ Kacheln, in denen alle Messwerte abgelegt werden. In der darüber liegenden Ebene gibt es nur ein Viertel der Kacheln und die Anzahl der zu speichernden Messwerte sinkt, da jede Kachel ein virtuelles Punkteraster (200×200) aufspannt und nur der dem Rasterpunkt nächstgelegene „reale“ Messwert abgespeichert wird (also maximal 40.000 Messwerte pro Kachel). Die Anzahl der Ebenen hängt von der Anzahl und Anordnung der Messwerte ab. Nach Anwendung dieses Verfahrens nimmt die Kachelanzahl von der unteren zur oberen Ebene hin ab. Sinnbildlich können diese Kachelebenen als Pyramide betrachtet werden. Trotzdem wird auf jeder Ebene der gesamte Messbereich abgebildet, da sich die Kantenlängen der Würfel in jeder Ebene verdoppeln (**Bild 5**).

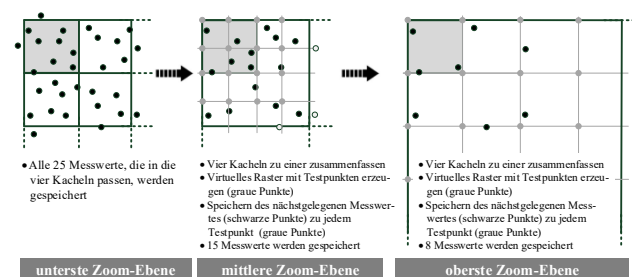


Bild 5 Verfahren zur Reduzierung von Messwerten für die unterschiedlichen Zoom-Ebenen

Für die Visualisierung wird die Zoom-Ebene 0 beim Öffnen einer Datei vollständig geladen. Beim Verschieben und Zoomen durch den Anwender wird die aktuelle Position des Fensterausschnitts ermittelt und über die Anzahl der sichtbaren Pixel (Monitorauflösung/Größe der Anwendung in Pixel) können die Kacheln berechnet werden, die nachzuladen sind (**Bild 6**). Für eine Visualisierung sind nur wenige Millionen Messwerte im Speicher zu halten und der Anwender kann sich nahezu verzögerungsfrei durch das Messvolumen bewegen (Abhängig von der Geschwindigkeit des Speichermediums/Netzwerks).

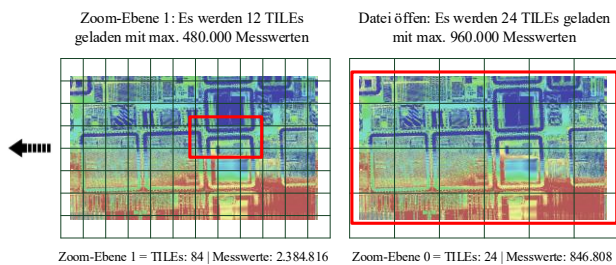


Bild 6 Größe der zu ladenden Daten beim Verschieben und Zoomen durch den Anwender (rotes Viereck)

Vorteil dieses Dateiformats ist das schnelle Laden der Messwerte, unabhängig von deren Anzahl. Als Nachteile sind der zusätzliche Zeitaufwand für die Konvertierung zu sehen und die relativ großen Dateien aufgrund der Redundanz von Messwerten in den unterschiedlichen Zoom-Ebenen. Die konvertierte Datei im TILE-Format hat ca. die 1,3-fache Größe einer nativen Messwertdatei.

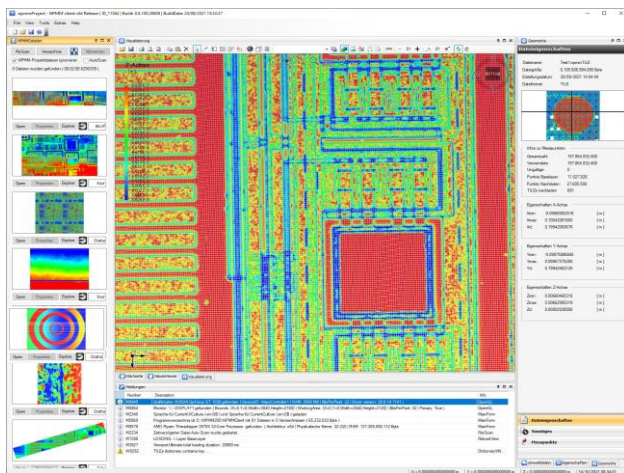


Bild 7 NPM200Client-Software zur Visualisierung von Messwertdateien im konvertierten TILE-Format

Der TILE-Format-Konverter und das Programm zur Visualisierung der Messwertdateien wurden über die Jahre ständig weiterentwickelt. Gestartet mit Standard-Rechner-Hardware wurden schnell die Grenzen erreicht, an denen ein effektives Arbeiten mit den entwickelten Programmen möglich war.

Die 2012 zur Verfügung stehende Rechentechnik mit INTEL-i7 Quad-Core CPU, 4 GB Arbeitsspeicher, 16 TB

RAID-Festplattenspeicher (konventionelle Festplattenlaufwerke) und einem kleinen Halbleiterlaufwerk (SSD) erreichte beim Konvertieren von 230 Mrd. Messpunkten und einem mittleren Punkteabstand von ca. 466 nm (ca. 5,5 TB große Datei) folgende Bearbeitungszeiten:

	2012	2013	2018	2020
CPU	i7-2600K	i7-2600K	i7-7700K	i9-9900K
RAM [GB]	4	32	64	64
HDD [TB]	16 - RAID	16 - RAID	16	0
SSD [TB]	0,5	0,5	4	16 - RAID
	keine Parallelisierung		teilweise parallelisiert	
Laufzeit	11d 18h	02d 16h	01d 22h	00d 23h

Tabelle 1 Programmlaufzeiten bei der Konvertierung mit dem TILE-Format-Konverter

Kleinere Optimierungen an Hard- und Software in den folgenden Jahren verbesserten die Konvertierungszeiten (**Tabelle 1**).

Der TILE-Format-Konverter verarbeitet die ca. 55.000 Aufnahmen der Kamera in folgenden vier Arbeitsschritten:

1. Eigenschaften der Messwertdateien bestimmen
2. Die Kacheln der untersten Zoom-Ebene mit allen Messwerten erzeugen
3. Die Pyramide der Zoom-Ebenen erzeugen, d.h. aus vier Kacheln eine Kachel für die darüber liegende Zoom-Ebene generieren
4. Alle 6,5 Millionen Kacheln in die Ziel-Datei schreiben und den Datei-Header aktualisieren

Eine Verbesserung der Laufzeiten des TILE-Format-Konverters durch vollständige Parallelisierung wurde auf einen späteren Zeitpunkt verschoben, da die begrenzenden Faktoren die Speichermedien (HDDs) waren bzw. bei Dauereinsatz die SSDs auf Grund steigender Temperaturen die Geschwindigkeiten stark reduziert hatten. Die CPU Auslastung lag unterhalb von 50% bis 60% und eine Parallelisierung hätte nicht den gewünschten Erfolg gebracht, da die Speichermedien bei einer Auslastung von 100% lagen.

5 Optimierung der Software

5.1 Auswahl neuer Hardwarekomponenten

5.1.1 Speichermedien

Ende 2020 waren die ersten bezahlbaren und großen U.2 Enterprise SSDs auch im Consumer-Bereich verfügbar und eine vollständige Parallelisierung der Programme wurde realisiert. Dabei wurde festgestellt, dass die Arbeitsschritte 1 und 3 gut und die Arbeitsschritte 2 und 4 mit Einschränkungen parallelisierbar sind. Durch den Einsatz einer U.2-SSD vom Typ MICRON 9300 PRO (16 TB) konnte mit

der neuen Programmversion die Bearbeitungszeit von ca. 23 Stunden auf ca. 18 Stunden reduziert werden. Die neuen Enterprise SSDs zeichnen sich dabei durch folgende Eigenschaften aus:

- Konstantes Lesen und Schreiben von bis zu 3,5 GB/s sequenzieller Daten
- Hohe Anzahl an Eingabe-/Ausgabeoperationen (4K Random Read IOPS 850000)
- NVMe Schnittstelle und Aluminium-Gehäuse (für hohe Leistung, auch unter thermischer Last)

5.1.2 Hauptprozessor (Mikroprozessor)

Nach erfolgreichem Austausch der Speichermedien und der Parallelisierung der vier Arbeitsschritte kehrt sich das Verhältnis der Auslastung der Rechenkomponenten um.

- Auslastung der CPU (INTEL i9-9900K) in den Schritten 1 und 3 nahezu 100%
- Auslastung der CPU (INTEL i9-9900K) in den Schritten 2 und 4 stark schwankend zwischen 30% und 80%
- Auslastung des Speichermediums (MICRON 9300 PRO) in den Schritten 1 und 3 kleiner 40%
- Auslastung des Speichermediums (MICRON 9300 PRO) in den Schritten 2 und 4 stark schwankend

Mit dem Einsatz einer 32-Kern CPU vom Typ AMD Threadripper™ 3970X und 128GB Arbeitsspeicher war die Erwartung einer Reduzierung der Konvertierungszeit mit dem TILE-Format-Konverter auf ca. $\frac{1}{4}$ der bisherigen 23 Stunden verbunden. Die ersten Tests lieferten folgende Ergebnisse:

- INTEL i9-9900K mit 128GB - ca. 23 Stunden
- AMD 3970X mit 128GB - ca. 54 Stunden

Die Erwartungen an eine Verringerung der Bearbeitungszeit konnten nicht erfüllt werden. In den folgenden Abschnitten wird daher auf Fehlersuche, Ursachenforschung und die daraus abgeleiteten Maßnahmen näher eingegangen.

5.2 Vergleichstest mit kommerzieller und nichtkommerzieller Software

Um genauere Aussagen über die Leistungsfähigkeit einer 2000€ teuren 32-Kern AMD-CPU machen zu können, wurden zunächst einige Vergleichstests mit vorhandenen Softwarepaketen durchgeführt. CPU-Benchmarks wurden nicht berücksichtigt, diese bilden aus unserer Sicht in vielen Fällen nicht die Anforderungen an unsere spezielle Software bei der täglichen Arbeit ab.

Mit einer professionellen Videoschnittsoftware wurde der erste Test durchgeführt. Ein ca. 45 Minuten langes Full-HD Video (max. 6 Videospuren) mit einer Datenrate von ca. 20 Mbit und Stereoton wurde gerendert. Dabei besteht jeder codierte Film, beziehungsweise Videostrom, aus aufeinanderfolgenden Bildergruppen. Diese Bildergruppen

beginnen stets mit einem Vollbild, auf das eine beliebige Anzahl Bilder mit Differenz-Informationen folgen, also ein sehr gut parallelisierbarer Prozess. Im Folgenden sind die Ergebnisse mit identischen Arbeitsspeicherbestückungen und Speichermedien (SSD) dargestellt:

CPU	Threads	FPS	Laufzeit
INTEL i7-12700K	20	53,12	00:53:25
INTEL i7-11700K*	16	50,25	00:56:34
INTEL i7-1065G7*	8	41,82	01:07:59
AMD 3970X	32	35,44	01:20:12
AMD 3970X	64	28,51	01:40:51
INTEL i9-9900K	16	24,50	01:56:02

Tabelle 2 Programmlaufzeiten beim Video-Rendering eines FULL-HD Videos (* unterstützt AVX512)

Ein weiterer Test erfolgte mit einer selbstgeschriebenen, multi-core fähigen Anwendung zur Berechnung des Reflexionsgrades von optischen Mehrfachschichten. Die AMD Threadripper™ 3970X CPU konnte hier nicht überzeugen.

CPU	Threads	Laufzeit	Laufzeit
INTEL i9-10900K	16	1104 sec	00:18:24
INTEL i9-9900K	16	1325 sec	00:22:05
INTEL i7-11700K*	16	1355 sec	00:22:35
INTEL i7-12700K	20	1475 sec	00:24:35
AMD 3970X	32	2763 sec	00:46:03
INTEL i7-1065G7*	8	3820 sec	00:63:40

Tabelle 3 Programmlaufzeiten im Programm Strahlverfolgung (* unterstützt AVX512)

Die Untersuchungen haben gezeigt, dass die 32-Kerne der AMD Threadripper™ 3970X CPU ihre Leistung nicht voll entfalten können. Es ist außerdem zu erkennen, dass immer die Kombination aus CPU und Software über die Geschwindigkeit entscheidet. Diese Ergebnisse konnten auch mit ANSYS verifiziert werden. Die Hoffnung, die Bearbeitungszeit auf ca. 25% zu verringern, indem die Anzahl der CPU-Kerne von 8 auf 32 erhöht wurde, konnte nicht erfüllt werden.

5.3 Unterschiede in Aufbau und Architektur von INTEL- und AMD-CPU's

Während AMD seit den ersten Ryzen™- bzw. Threadripper™-Prozessoren hauptsächlich Chiplets verwendet, setzt Intel vorwiegend auf monolithische Chips (alle Komponenten auf einem Die). Wenn sich beim monolithischen Design die Fläche z.B. bei erhöhter Kernanzahl vergrößert, können sich leichter Fehler einschleichen und das Exemplar wird zum Ausschuss oder zweitklassiger Ware. Durch die Aufspaltung eines monolithischen Designs in

Chiplelets kann die Ausbeute erhöht werden, zudem kann die optimale Strukturbreite für unterschiedliche Funktionsmodule gewählt werden. Chiplelets lassen sich für andere Chips wiederverwenden und mehrere überschaubare Chiplelets lassen sich einfacher modernisieren als eine monolithische Gesamtlösung. Andererseits sind IO-Module notwendig, über die die einzelnen Chiplelets miteinander kommunizieren. Das führt aber auch zu neuen Problemen:

- Die Latenzen zwischen den Kernen (INTEL i7-1065G7 oder INTEL i9-10900K) liegt bei ca. 40 ns und bei den AMD-CPU's bei ca. 100 ns. Grund dafür ist der komplexere Aufbau der AMD-CPU's aus 8 Quad-Core-Chiplelets im Vergleich zu einer Kommunikation über einen Ring-Bus bei monolithischen INTEL-CPU's, welche keine Chiplelets verwenden [4].
- Der INTEL Core i9-10900K (10-Kerner) benötigt 25 ns (andere Quellen: 50 ns bis 54 ns) um auf den Arbeitsspeicher zuzugreifen, der AMD Ryzen™ 3950X (16-Kerner) braucht dafür mit 59 ns mehr als die doppelte Zeit. Beim 32-Kerner AMD Threadripper™ 3970X ist das Verhältnis noch schlechter (77 ns bis 104 ns). Bei beiden CPU-Typen hängt das allerdings auch vom verwendeten Speichertyp ab [5].
- Die All-Core-Speed der INTEL i9-10900K liegt bei ca. 4,9 GHz und der AMD Threadripper™ 3970X bei ca. 3,8 GHz.

Damit sich unterschiedliche Latenzen nicht negativ auf die Performance auswirken, müssen Anwendungen und der Scheduler des Betriebssystems an die spezielle Prozessorarchitektur angepasst sein, was Windows 10 erst seit der Version 1903 kann [4].

5.3.1 Maßnahmen zur Verringerung des Einflusses der CPU-Architektur

Um die höheren Latenzen bei der Chiplet-Architektur (u.a. Kommunikation über IO-Module bzw. Bussystem) zu kompensieren, wurden die Aufgaben bei der Konvertierung der Messdaten neu partitioniert. Es wurde versucht, die Abhängigkeiten einzelner Arbeitsschritte weitestgehend zu minimieren, so dass die Chiplelets möglichst wenig über das Bussystem miteinander kommunizieren müssen. Auch parallele Zugriffe auf Dateien (mit File-Locking) wurden eliminiert. Durch Trial-and-Error wurde nach vielen Tests die neue Aufteilung der Konvertierungsaufgabe in einzelne Arbeitsschritte definiert:

1. Ermitteln der Anzahl der Messwerte
2. Berechnen der Eigenschaften der Messwerte
3. Erzeugen aller Kacheln für die unterste Kachel-Ebene (einige zusammengehörende Kacheln sind noch in mehreren Dateien aufgeteilt)
4. Zusammenführen geteilter Kacheln in eine Kachel
5. Verschieben aller Kacheln in gemeinsamen Ordner
6. Erzeugen weiterer Zoom-Ebenen
7. Kopieren der temp. Kacheldateien in die Zielfeile

Die Arbeitsschritte eins bis sechs sind gut parallelisierbar, der Schritt sieben ist nicht parallelisierbar. Schritt sechs und Schritt sieben starten parallel, um die im Überfluss vorhandenen Kerne/Threads der AMD-CPU sinnvoll auszulasten. Anhand von detaillierten Bearbeitungszeiten ist erkennbar, dass das Erzeugen weiterer Zoom-Ebenen noch nicht optimal läuft. Obwohl jetzt mehr Arbeitsschritte auszuführen sind, konnten bei der AMD-CPU schon signifikante Verbesserungen erreicht werden (**Tabelle 4**).

CPU	Cores/Threads	12/2020	01/2021
INTEL i9-9900K	8/16	ca. 23 h	22:57:17
AMD 3970X	32/32	ca. 54 h	25:29:26

Tabelle 4 Programmlaufzeiten nach Optimierung der Programmstruktur und Einführung der sieben Arbeitsschritte

Die Optimierung der Programmstruktur führte bei der AMD Threadripper™ 3970X CPU zu mehr als einer Halbierung der Bearbeitungszeit. Die INTEL-CPU profitiert von den durchgeführten Änderungen geringfügig, was aber trotzdem auf eine suboptimale Programmstruktur bzgl. der Parallelität in der ursprünglichen Version hindeutet.

Interessant ist auch das Verhalten und die Auswirkungen einer unterschiedlichen Anzahl von Kernen/Threads auf die Bearbeitungsdauer von Programmen mit den AMD- bzw. INTEL-CPU's. Der Test erfolgte aus Zeitgründen mit einer reduzierten Anzahl von Messwerten (2,8 Mrd. Messwerte und einem mittleren Punkteabstand von 626 nm) sowie dem Abschalten von Kernen/HyperThreading/Simultaneous Multithreading über das BIOS (**Tabelle 5**).

CPU / THREADS	64	32	24	16	8
AMD 3970X	758 s	710 s	677 s	696 s	764 s
INTEL i9-9900K				550 s	573 s

Tabelle 5 Programmlaufzeiten in Abhängigkeit der verwendeten Kerne/Threads

HyperThreading (INTEL) und Simultaneous Multithreading (AMD) bezeichnet die Fähigkeit eines Mikroprozessors, mittels getrennter Pipelines und/oder zusätzlicher Registersätze mehrere Threads gleichzeitig auszuführen. Wie in [6] beschrieben, gibt es Anwendungsszenarien, bei denen das eingeschaltete HyperThreading/Simultaneous Multithreading kontraproduktiv ist und die Abarbeitungszeiten verlängert. Im Falle der AMD-CPU mit ihren 32 Kernen scheint das Umschalten der Pipelines und zusätzlichen Registersätze mehr Zeit zu beanspruchen, als nur mit „echten“ Kernen zu rechnen. Dieses Verhalten ist sicherlich anwendungsbezogen und muss individuell definiert werden. Im Falle der Anwendung der NPM200 wird die AMD Threadripper™ 3970X CPU ohne Simultaneous Multithreading und die INTEL i9-9900K mit HyperThreading verwendet.

5.3.2 C# Funktionen und deren Timings

Die Anwendung für die NPMM200 wurde in C# mit dem .NET-Framework (Version 4.7.2) und Microsoft Visual Studio 2019 unter Windows entwickelt. Bereits in Version 3.5 des .NET-Frameworks standen 11.400 Klassen zur Verfügung [7]. Andere Quellen sprechen von bis zu 14.000 Klassen für aktuelle Versionen des .NET-Frameworks. So ist es nicht verwunderlich, dass für eine Aufgabe mehrere Klassen Verwendung finden können.

Unter diesem Aspekt wurde der sechste Bearbeitungsschritt weiter optimiert. Die AMD-CPU hat hier eine Auslastung von nur ca. 30%. Zur Erinnerung: Der sechste Verarbeitungsschritt macht aus vier Kacheln eine neue Kachel (mit ca. einem Viertel der Messpunkte) für die darüber liegende Zoom-Ebene. Eigentlich sehr gut parallelisierbar, aber in Realität der Verarbeitungsschritt, der die meiste Zeit benötigt.

Der Programmcode für diesen Verarbeitungsschritt wurde umfassend mit „digitalen Stoppuhren“ versehen und alle relevanten Programmabschnitte wurden bezüglich des Timings überprüft. Die C#-Stopwatch-Klasse hat eine Auflösung von 15 ns (entspricht 1 Tick). Folgende Programmabschnitte wurden als problematisch detektiert:

- Das Umwandeln der aus Dateien eingelesenen Bytefelder in DOUBLE-Werte
- Das Erzeugen der Instanzen der 3D-Point (XYZ) Objekte aus den DOUBLE-Werten
- Das Lesen der temporären Dateien beim Wegschreiben in die Zielfeile

Als großer Ressourcenverbraucher hat sich das Erzeugen von Instanzen der 3D-Point-Klasse aus DOUBLE-Werten herausgestellt. Dass das Erzeugen von Instanzen zeitaufwendig ist, war bekannt. Neben der Speicherzuweisung für die Objekte der Klasse müssen auch die META-Daten der Objekte angelegt und initialisiert werden. Ein normales C#-Programm muss einige Duzend, vielleicht auch wenige tausend Objekte pro Sekunde instanzieren. In unserem Fall sprechen wir von vielen hundert Millionen 3D-Punkten, die für die Berechnungen erzeugt werden müssen. Dieser Zeitaufwand wurde eindeutig unterschätzt.

Obwohl Klassen viele Vorteile bieten, in unserem Fall die eingebauten Funktionen zur Berechnung von Punkteabständen im 2D- und 3D-Raum, wurde das Programm so geändert, dass anstelle von Klassen einfachere Strukturen verwendet werden. Als Programmierer ist der Verwaltungsaufwand beim Einsatz von Strukturen größer, da die CLR (Common Language Runtime) des .NET-Frameworks sich nicht um Speichervorgaben und Speicherfreigabe kümmert. Als Ergebnis dieser Änderungen konnte eine signifikante Zeitersparnis erreicht werden.

Ein weiterer Ressourcenverbraucher ist das Umwandeln der aus den ca. 4 MB großen Messwertdateien gelesenen Bytefelder in DOUBLE-Werte für die 3D-Point-Strukturen (früher der 3D-Point-Objekte). Auch dafür stehen mehrere Klassen und Funktionen (z.B. BinaryReader.ReadDouble(), BitConverter.ToDouble() usw.) zur Verfügung.

Tests haben gezeigt, dass am schnellsten das vollständige Einlesen der temporären Dateien mit BinaryReader.ReadBytes() in ein Bytefeld und das anschließende Auslesen der DOUBLE-Werte mit Zeigern und UNSAFE-Code funktioniert. Auch UNSAFE-Code sollte als C#-Programmierer vermieden werden, denn dieser Code läuft außerhalb der verwalteten Umgebung des .NET-Frameworks und wird nicht von der CLR erfasst. Damit verzichtet man auf die Debug Engine, den Garbage Collector, das Thread Management und viele andere Funktionen für die Ausführung von C#-Programmen. Auch das Aufräumen des Arbeitsspeichers muss der C#-Programmierer händisch implementieren. Trotz aller Nachteile von UNSAFE-Code konnte auch hier eine signifikante Zeiteinsparung erreicht werden.

Weitere Optimierungsarbeiten betrafen den Umgang mit temporären Dateien und der Zielfeile. Auch hier bietet das .NET-Framework eine Vielzahl von Klassen und Funktionen an (z.B. FileStream, File, BinaryReader usw.). Über Parameter wie FileShare, AccessMode, BufferSize oder ScanMode können die Klassen und Funktionen zum Dateihandling detailliert gesteuert werden. Über Zeitmessungen und die Untersuchung einer Vielzahl an Kombinationen konnte die Klasse FILE als die schnellste Klasse ermittelt werden. Um eine bessere Vorstellung zu bekommen, hier die Anzahl an Dateien, die mehrfach geschrieben, gelesen oder gelöscht werden müssen, am Beispiel eines Übersichtsbildes für den gesamten Messbereich (**Tabelle 6**).

Zoom-Ebene	Anzahl der Dateien	Anzahl der 3D-Punkte
1	4.947.827	197.864.832.400
2	1.238.112	49.488.215.400
3	310.106	12.286.365.576
4	77.796	3.069.068.716
5	19.589	770.481.523
6	4.937	192.496.416
7	1.296	48.110.233
8	324	12.037.937
Gesamt	6.599.987	263.731.608.201

Tabelle 6 Anzahl von Dateien bzgl. Zoom-Ebene während des Konvertierens in das TILE-Format

Erste Messungen haben gezeigt, dass beim Öffnen und Einlesen der ca. 4 MB großen Dateien nur ca. 400 Dateien pro Sekunde verarbeitet werden konnten. Durch Verwendung der viel schnelleren Klasse FILE ist es jetzt möglich, ca. 800 bis 1000 Dateien pro Sekunde lesend bzw. schreibend zu verarbeiten, was ungefähr einer Halbierung der bisher benötigten Bearbeitungszeit entspricht. Mit diesen Werten erreicht man speziell in den Bearbeitungsschritten 1, 2, 3 und 6 die maximal möglichen Schreib- bzw. Lesegeschwindigkeit der MICORN 9300 PRO U.2 SSD von ca. 3.500 MByte/s. Eine bessere Performance ist beim Einsatz derartiger Speichermedien im RAID-Verbund möglich.

5.3.3 Zusammenfassung der Ergebnisse der Optimierung von Programmarchitektur und Software

Die Untersuchungen haben gezeigt, dass sich die Ausführungsgeschwindigkeiten von Funktionen auf INTEL- bzw. AMD-CPU unterscheiden. Am deutlichsten wird das bei der Umwandlung von Bytefolgen in DOUBLE-Werte bzw. umgekehrt. Hier wurden vor allem Schwächen bei der AMD-CPU ermittelt. Bei schlecht parallelisierbaren Arbeitsschritten, die nach dem Prinzip Producer-Consumer arbeiten, hat die INTEL-CPU aufgrund der hohen Single-Core-Speed Vorteile. Gut parallelisierbare Arbeitsschritte, mit wenig Speicher- oder Festplattenzugriffen sind eine Domäne der AMD-CPU. In **Tabelle 7** sind die Ergebnisse der Programmlaufzeiten nach der Optimierung dargestellt.

Anzahl	198 Mrd. Messwerte			
Messbereich	198x198 mm			
Abstand	466 µm Pixel zu Pixel			
Pixel / TILE	39.990			
Anzahl Tiles	4.947.827			
Dateigröße	6.323 GB			
CPU	INTEL i9-9900K		AMD 3970X	
Version	03/2021	08/2021	03/2021	08/2021
Schritt 1	00:07:21	00:07:23	00:02:41	00:02:34
Schritt 2	00:05:32	00:05:27	00:01:37	00:01:46
Schritt 3	02:24:48	01:57:46	01:37:32	01:45:55
Schritt 4	00:24:54	01:18:24	00:47:57	00:52:19
Schritt 5	00:11:55	00:10:16	00:12:30	00:12:02
Schritt 6	18:26:26	04:57:04	22:30:39	03:50:57
Schritt 7	05:06:30	02:42:26	11:19:49	02:01:06
Gesamt	22:57:17	08:49:36	25:29:26	07:00:58

Tabelle 7 Programmlaufzeiten nach Optimierung

Durch Änderungen an Programmstruktur und Austausch einzelner Klassen und Funktionen konnte eine signifikante Zeitersparnis bei der Konvertierung der Messwertdateien einer NPM200 erzielt werden. Noch bessere Laufzeiten können erreicht werden, wenn im Programmcode an ausgewählten Stellen CPU-spezifische Funktionen eingesetzt werden. Diese Möglichkeiten wurden aktuell nicht weiterverfolgt, da zu Testzwecken nicht genügend viele unterschiedliche AMD-CPU zur Verfügung standen und keine stichhaltigen Aussagen dazu gemacht werden können. Gleichzeitig steigt der Aufwand bei Pflege und Weiterentwicklung der Software, auch Microcode Updates der CPU-Hersteller können zu anderen Timings oder dem Abschalten spezieller Funktionen führen (geschehen bei INTEL i7- und i9-CPU der 12-ten Generation mit AVX512 [8]). Aus diesen Gründen gibt es nur eine Programmversion die beiden CPU-Architekturen gleichermaßen gerecht wird.

5.4 Arbeitsspeicher und Speichermedien

5.4.1 Kurze Betrachtung zum Arbeitsspeicher

Gestartet 2012 mit 4 GB Arbeitsspeicher lag ein Großteil der Last auf mechanischen Festplatten (HDDs) mit ihren schlechten Zugriffszeiten und noch geringen Schreibraten. Beim Wechsel von 4 GB auf 16 GB und später (2018) auf 64 GB konnte die Abarbeitungszeit stark verringert werden (von ca. 13 Tagen auf ca. 2 Tage). Ob 128 GB Arbeitsspeicher auf dem INTEL-i9-9900K Rechner die Verarbeitungszeit weiter verkürzen würden, kann leider nicht getestet werden, da das Mainboard nur maximal 64 GB Arbeitsspeicher unterstützt. Die AMD Threadripper™ 3970X CPU, einschließlich Mainboard, unterstützt bis zu 256 GB Arbeitsspeicher, aber auf Grund mechanischer Inkompatibilitäten zwischen CPU, Wärmetauscher der Wasserkühlung und den RAM-Modulen wird diese nur in einer Ausbaustufe mit 128 GB Arbeitsspeicher betrieben.

Trotzdem wurden Tests durchgeführt, die den Einfluss des Arbeitsspeichers bei der Datenkonvertierung untersuchen. Die Programmarchitektur ist so angelegt, dass möglichst viele Daten im Arbeitsspeicher gehalten werden. Ist kein Arbeitsspeicher mehr verfügbar, wird „aufgeräumt“ und alle temporären Daten werden in Dateien gespeichert. Damit entlastet natürlich mehr Arbeitsspeicher selbst die heutzutage relativ schnellen NVMe SSDs signifikant.

- INTEL-CPU (64 GB Arbeitsspeicher) - 195 Mal müssen alle offenen Dateien geschlossen werden
- AMD-CPU (224 GB Arbeitsspeicher) – 35 Mal müssen alle offenen Dateien geschlossen werden

Am Beispiel einer INTEL i7-12700K CPU (SSD MICRON PRO 9300 | 15,36TB | PCIe-U.2-SSD | TLC) wurden unterschiedliche Speicherkonfigurationen auf die Programmlaufzeiten getestet (**Tabelle 8**).

	32 GB	64 GB	96 GB	128 GB
Schritt 1	00:03:08	00:03:07	00:03:18	00:03:09
Schritt 2	00:02:24	00:02:25	00:02:37	00:02:13
Schritt 3	02:56:01	01:43:55	01:49:42	01:32:16
Schritt 4	01:15:20	01:20:16	01:22:37	01:17:41
Schritt 5	00:05:57	00:05:56	00:05:47	00:05:40
Schritt 6	06:18:47	04:51:59	04:19:20	04:01:16
Schritt 7	03:34:26	04:08:41	03:33:32	02:59:13
Gesamt	10:49:34	08:22:42	07:57:53	07:10:05

Tabelle 8 Programmlaufzeiten bzgl. Speicherausstattung

INTEL-Consumer CPUs unterstützen aktuell maximal 128 GB Arbeitsspeicher, was auch unter finanziellen Gesichtspunkten (ca. 600€) eine sinnvolle Speicherausstattung für das NPM200Client-Programm darstellt.

5.4.2 Kurze Betrachtung zu Speichermedien

Großen Einfluss bei der „Datenverarbeitung“ hat natürlich die Geschwindigkeit und Größe der Speichermedien. Gestartet mit herkömmlichen Festplatten (HDDs) und einer maximalen Übertragungsgeschwindigkeit von ca. 100 MB/s sind wir 2021 bei PCIe-SSDs mit einer maximalen Übertragungsgeschwindigkeit von bis zu 3.500 MB/s angekommen. Dass diese hohen Geschwindigkeiten nicht immer erreicht werden, steht auf einem anderen Blatt. Die Hersteller sehen nur den „normalen“ Endverbraucher mit Internetsurfen und Office, ansonsten ist der Preis entscheidend. Das große (und preiswerte) NVME-SSDs nicht das halten was die Werbung verspricht, sehen wir sehr eindrucksvoll am Beispiel der Corsair MP400. Anwender mit höheren Ansprüchen sollten, resultierend aus dem Test, lieber zu einer Enterprise PCIe-U.2-SSD mit TLC (Triple Level Cells) greifen.

Für die durchgeführten Tests standen eine Vielzahl an Speichermedien zur Verfügung. Für die Tests wurden diese immer zwischen den beiden Rechnerplattformen gewechselt (ebenso die Module des Arbeitsspeichers)

1. Seagate IronWolf 10000,8 GB (SATA-HDD)
2. Corsair MP400 4000,7 GB (PCIe-M.2-SSD | QLC)
3. Samsung SSD 860 EVO 4TB (SATA-SSD | TLC)
4. Samsung SSD 970 EVO 2TB (M.2-SSD | TLC)
5. MICRON 9300 Pro 15362,9 GB (U.2-SSD | TLC)

Nur die MICRON 9300 PRO ist in der Lage, die Daten des kompletten Messbereichs von 200x200 mm zu verarbeiten. Aus diesem Grund wurde für alle Speichermedien ein Datensatz von 26 Mrd. Messwerten für die Untersuchungen verwendet.

	Platte 1	Platte 2	Platte 3	Platte 4	Platte 5
1	00:05:50	00:00:59	00:01:06	00:00:59	00:01:03
2	00:00:39	00:00:42	00:00:42	00:01:02	00:00:42
3	01:17:56	00:25:40	00:24:11	00:16:14	00:11:31
4	01:39:37	00:10:13	00:12:37	00:07:44	00:02:52
5	00:04:29	00:02:06	00:00:45	00:00:42	00:00:37
6	06:02:01	01:11:41	01:04:43	00:59:25	00:59:30
7	06:17:16	00:39:55	00:15:00	00:18:24	00:18:27
G	12:41:47	02:42:58	01:54:02	01:27:57	01:17:38

Tabelle 9 Programmlaufzeiten bzgl. Speichermedium

Das langsamste SSD-Speichermedium benötigt mehr als die doppelte Zeit des schnellsten SSD-Speichermediums. Berücksichtigt man den Preis der SSDs, der bei allen Laufwerken ähnlich ist (2.100€ bis 2.600€ für 16 TB), macht es durchaus Sinn, Laufwerke auszuwählen, die über lange Zeit ihre Leistungsparameter gewährleisten. Bei der Verarbeitung der Messwertdateien sind mit den untersuchten Speichermedien folgende Besonderheiten aufgetreten:

- Alle SSDs wurden zu warm und erreichten bei der Abarbeitung bestimmter Arbeitsschritte eine Auslastung von 100%
 - ❖ Samsung SSD 970 - 80 °C
 - ❖ MICRON 9300 Pro - 75 °C
 - ❖ Corsair MP400 - 69 °C
 - ❖ Samsung SSD 860 - 59 °C
 - ❖ Seagate IronWolf - 57 °C
- In Grenzsituationen haben die Corsair MP400 und die Samsung SSD 970 EVO sowohl den INTEL-Rechner als auch den AMD-Rechner zum Absturz gebracht. Symptome waren das Einfrieren aller Anwendungen und ein sich anschließender, automatischer Neustart der Rechner. Dieser Effekt trat mehrfach auf, ob andere Gründe dieses Verhalten hervorgerufen haben, kann nicht ausgeschlossen werden. Hohe Temperaturen der Speichermedien und Abstürze lagen zeitlich dicht beieinander.
- Bei steigender Temperatur drosseln die PCIe-SSDs die Lese- und Schreibgeschwindigkeiten, im Falle der Corsair MP400 (**Bild 8**) bis auf magere 85 MB/s (laut Datenblatt 3.400 MB/s für sequenzielles Lesen und 3.000 MB/s für sequenzielles Schreiben).
- Einzig die MICRON 9300 PRO hat unter allen Bedingungen eine Bearbeitung der Daten zugelassen, wenn auch mit gedrosselter Geschwindigkeit und Temperaturen im roten Bereich. Trotzdem wurden noch weit über 1.500 MB/s übertragen (laut Datenblatt 3.500 MB/s für sequenzielles Lesen und 3.500 MB/s für sequenzielles Schreiben)

Grund für den starken Einbruch der Lese- bzw. Schreibleistung bei SSD-Speichermedien sind die verwendeten Speicherbausteine. NAND-Flash vom Typ QLC (Corsair MP400) speichert vier Bit pro Speicherzelle und sorgt so für niedrigere Kosten, ist aber in puncto Leistung und Haltbarkeit schlechter aufgestellt als TLC-3D-NAND mit drei Bit pro Zelle.

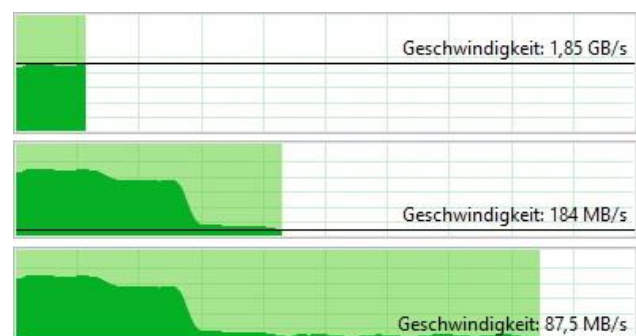


Bild 8 Kopiervorgang der Corsair MP400 (140 GB) und das Verhalten von QLC-Speicher (Quadruple Level Cells)

Aufgrund des vorhandenen SLC-Caches werden bei der Corsair MP400 die ersten Gigabytes mit deutlich über 1 GB/s geschrieben, danach fällt die Leistung auf minimal

80 bis maximal 200 MB/s ab, bis die Daten dann wirklich mit 4 Bit pro Speicherzelle geschrieben werden.

Im Vergleich dazu das Verhalten von HDDs und SSDs (mit TLC-Speicher) beim Kopieren der 140 GB großen Testdatei (**Bild 9**).



Bild 9 Kopiervorgang mit 140 GB großer Testdatei

Im Dauerbetrieb ist aus diesem Grund eine konventionelle HDD von der Schreibleistung einer SSD mit QLC-NAND leicht überlegen (jedenfalls beim Schreiben sequenzieller Daten). Beim Lesen oder Schreiben zufälliger Daten können HDDs mit Zugriffszeiten von ca. 10 ms im Vergleich zu SSDs mit 0,2 ms nicht mithalten.

Die nächste Generation an SSD-Speichermedien wird den neuen PCIe 4.0 unterstützen, mit dem sich die theoretischen Schreib- und Leseraten verdoppeln werden.

Beim Einsatz unter diesen speziellen Bedingungen sind PCIe-SSDs eher als „Verbrauchsmaterial“ zu betrachten, da sich hohe Temperaturen über einen längeren Zeitraum sicherlich negativ auf die Lebensdauer auswirken.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Beim Scannen einer Fläche von 198x198 mm (mit Wafer) wurden 125.625 Bilder aufgenommen, für das Übersichtsbild wurden 52.900 Bitmaps (zu je 3.745.302 Bytes) und das zugehörige NUMPY-Array (zu je 3.740.484 Bytes) abgespeichert. Die Ordner mit den Daten der Fokussuche und des Übersichtsbildes haben eine Größe von 889 GB und umfassen insgesamt 251.247 Dateien in 1.753 Ordnern. Die Dateien des Übersichtsbildes (*.bmp und *.npy) belegen 369 GB und müssen für eine Weiterverarbeitung aufgehoben werden. Nach der Konvertierung in das TILE-Format hat die Zieldatei eine Größe von 5,62 TB (**Bild 10**).

Der ursprüngliche Ansatz, die Rechenzeit durch bessere Hardware zu reduzieren, hat sich auf Grund unterschiedlicher Rechnerarchitekturen als nicht praktikabel erwiesen.

Die Untersuchungen haben gezeigt, dass die Software eine stärkere Anpassung an die unterschiedlichen Hardwarekomponenten benötigt, um deren Potential vollständig auszuschöpfen.

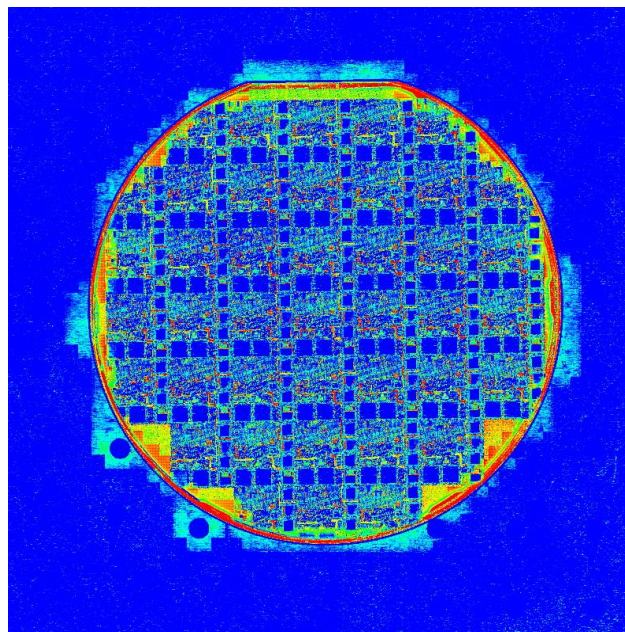


Bild 10 Übersichtsbild eines Wafers (6 Zoll) auf INVAR-Grundplatte

Der Aufwand zur Optimierung der Software war hoch, speziell bei der Ursachenforschung zur Ermittlung des Timings einzelner Funktionen, hat aber zu einer signifikanten Reduzierung der Bearbeitungszeit geführt (**Tabelle 10**).

CPU	Kerne	2020	2021
AMD 3970X	32	54:00:00	07:00:58
INTEL i9-9900K	8	23:00:00	08:49:36
INTEL i7-12700K	8 ^P + 4 ^E		07:10:05
INTEL i9-12900K	8 ^P + 8 ^E		???

Tabelle 10 Programmlaufzeiten nach Optimierung der NPM200-Software (^P Leistungskerne, ^E Effizienzkerne)

Keine Berücksichtigung fanden bisher die Hardwarekosten. Mit der Annahme, dass die Kosten für Mainboards, Arbeitsspeicher, Grafikkarte, Speichermedien und Gehäuse ähnlich sind, spielen nur die Kosten der CPU eine relevante Rolle (**Tabelle 11**).

CPU	Kerne	Preis 03/2022
AMD 3970X	32	2.299 €
INTEL i9-9900K	8	679 € ¹
INTEL i7-12700K	8 ^P + 4 ^E	409 €
INTEL i9-12900K	8 ^P + 8 ^E	589 €

Tabelle 11 Preise ausgewählter Prozessoren (¹ 01/2019)

Im konkreten Fall der Software für die NPMM200 bieten INTEL-Rechnersysteme ein sehr gutes Preis-/Leistungsverhältnis. Mit der Verfügbarkeit einer aktuellen INTEL i9-12900K CPU dürfte der Vorsprung der AMD Threadripper™ 3970X CPU weiter sinken. Erst dann wird auch deutlich, inwieweit jetzt schon die MICRON 9300 PRO SSD im U.2-Format die begrenzende Hardwarekomponente sein wird.

6.2 Ausblick

Obwohl die NPMM200 Messdaten und Bilder als Stream zur Verfügung stellen kann, wird diese Funktionalität aktuell nicht verwendet. Die Konvertierung wird manuell nach einer Messung über den im NPMM200Client-Programm integrierten TILE-Format-Konverter gestartet.

Die zukünftige Entwicklung wird sich verstärkt einer Client-Server-Architektur widmen, in der die Messdaten und Bilder an einen Server gestreamt werden, der dann, auch schon während der Messung, die Daten verarbeitet bzw. konvertiert. Die Speicherung der ins TILE-Format konvertierten Messdateien erfolgt ebenfalls auf diesem Server.

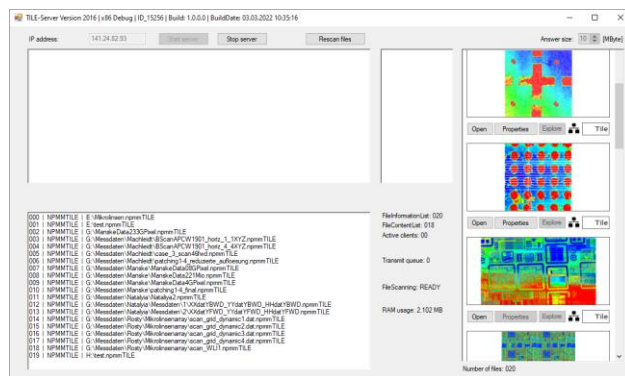


Bild 11 Prototyp der Serverkomponente (ohne Streaming-Funktionen und Konvertierung)

Prototypen der Serverkomponente sowie der Clientkomponente (als Modul des NPMM200Client-Programms) sind bereits vorhanden (**Bild 11**). Die Visualisierung erfolgt über das NPMM200Client-Programm, da nur sichtbare TILES in der entsprechenden Zoom-Ebene geladen werden müssen, ist dafür eine Gigabit-Netzwerkverbindung vollkommen ausreichend. Schnelle RAID-Systeme aus konventionellen HDDs werden als kostengünstige Variante der Datenspeicherung eingesetzt, während performante PCIe-SSD (U.2 oder U.3) für die Konvertierung unabdingbar sind.

7 Literaturverzeichnis

- [1] Weidenfeller, Laura: Kombination von zweiphotonen-basiertem direktem Laserschreiben mit großflächiger und hochpräziser Nanopositionierung. Ilmenau: Universitätsverlag Ilmenau, 2020, S. 75.
- [2] <https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system>. Aufgerufen: 25.02.2022.

- [3] McGovern, R., Building Your Own Tile Server, 07.06.2012, [https://docs.microsoft.com/en-us/previous-versions/bb545006\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/bb545006(v=msdn.10)?redirectedfrom=MSDN). Aufgerufen: 25.02.2022.
- [4] Hirsch, Christian: Um Antwort wird gebeten – Kern zu Kern Latenzen bei modernen Prozessoren. Hannover: Heise Medien, 2021, Ausgabe 2, S. 132.
- [5] Hirsch, Christian: Zehn bringt zehn – Core i9-10900K und Core i5-10600K gegen Ryzen™ 3000. Hannover: Heise Medien, 2022, Ausgabe 13, S. 86.
- [6] Vötter, Rafael: Ryzen™ 9 5950X: Leistungsbremse SMT? Benchmark-Vergleich 32 Threads vs. 16 Kerne mit Überraschungen. <https://www.pcgameshardware.de/Ryzen™-9-5950X-CPU-276916/Specials/SMT-Vergleich-CPU-Test-mit-Radeon-1364596>, Aufgerufen: 28.02.2022, Fürth: COMPUTEC MEDIA GMBH, 2021, Ausgabe 13, S. 86.
- [7] <https://de.wikipedia.org/wiki/.Net-Framework>, Aufgerufen: 01.02.2022
- [8] Windeck, Christof: Bit-Rauschen – SGX-Abschaltung, Zen-2-Revival und ARM-CPU mit AMD-GPU. Hannover: Heise Medien, 2022, Ausgabe 04, S. 40.