Konstruktion eines computergesteuerten Messstandes zur Untersuchung temperaturbedingter optischer Ablenkungen

Verena Hamburger
Juni 2002

Projektarbeit im Fachbereich Informatik der Universität Kaiserslautern

Autor: Verena Hamburger
Email: Verena@joschs-robotics.de

Betreuung: Prof. E. von Puttkamer
Anschrift: AG Robotik & Prozessrechentechnik
Postfach 3049
67653 Kaiserslautern

Inhalt

1. Einleitung und Motivation

Die vorliegende Arbeit beschreibt den Aufbau und die Steuerung eines computergesteuerten Temperaturmessstandes zur Untersuchung optischer Ablenkungen von Prismenverbänden bei thermischer Ausdehnung.

Thermische Effekte haben ein großen Einfluss auf Präzisionsmechaniken und –optiken jeder Art. Insbesondere bei hochgenauen Vermessungsgeräten ist es wichtig, dass die verwendeten Bauteile einen möglichst kleinen Temperaturausdehnungskoeffizienten aufweisen, da die Winkeländerung eines Laserstrahls ausgelöst durch die temperaturabhängige Ausdehnung der Versuchanordnung meist bereits in der gleichen Größenordnung liegt wie der zu messende Effekt.

Diese hohen Anforderungen an ein definiertes Temperaturverhalten werden auch an die von der Firma AndroTec GmbH zur mobilen Ortsbestimmung entwickelte RoboStation (siehe Abb. 1) gestellt. Zu den Haupteinsatzgebieten dieses Positionsbestimmungssystems zählen u.a. die Positionierung mobiler Serviceroboter und fahrerloser Transportsysteme, sowie die Bauvermessung und -ausführung.

Abb. 1: RoboStation im Einsatz

Da auch hier die verwendeten optischen Elemente z.T. sehr temperaturempfindlich sind, werden die Messergebnisse durch Temperaturschwankungen beeinflusst. Ohne eine Temperaturkompensation könnte es passieren, dass ein Roboter, der einen festen Punkt im Raum ansteuert, im Winter an einer anderen Stelle ankommt als im Sommer. Um diese

systematischen Messfehler bestmöglichst ausgleichen zu können, sollen im Rahmen dieser Projektarbeit die Grundlage zur Erfassung des Zusammenhangs zwischen Temperatur und Messergebnis mittels eines PC-gesteuerten Temperaturschranks geschaffen werden.

2. Messprinzip

Da das RoboStation-System seine Messungen mittels Laser und Spiegeln vornimmt, bewirken schon kleine Verschiebungen der Spiegel messbare Abweichungen vom korrekten Ergebnis. Solche kleine Verschiebungen treten auch bei Temperaturschwankungen auf, da Spiegel wie Halter sich bei Hitze ausdehnen und bei Kälte zusammenziehen. Dieses Kapitel beschreibt die Vorgehensweise dieser Projektarbeit um diese systematischen Abweichungen zu erfassen.

Grundsätzlich dient die im Rahmen dieses Projekts entwickelte Anordnung lediglich dazu den oben beschrieben Effekt messbar zu machen. Da bekannt ist, dass zwischen Ausdehnung und Temperaturänderung ein fester Zusammenhang besteht, der in gewissen Grenzen sogar annähernd linear verläuft, kann durch Kenntnis dieser Größen eine Korrekturfunktion erstellt werden. Die Messung der Ausdehnung geschieht hierbei indirekt über die Ablenkung eines einfallenden Laserstrahls, da die Ausdehnung der Optik proportional zur Winkeländerung des Lasers ist.

Zur Initialisierung erwärmt (bzw. kühlt) man zunächst das zu vermessende optische System in einem nach außen thermisch isolierten Behälter (im weiteren Temperaturschrank genannt) auf eine feste Temperatur. Sobald diese Bezugstemperatur in der gesamten Vorrichtung erreicht ist, wird ein Laser eingeschalten, dessen Strahl durch eine kleine Öffnung in der Außenwand auf die Messvorrichtung fällt, diese passiert und den Schrank auf der anderen Seite - aufgeteilt in drei Laserstrahlen - wieder verlässt. Die abgelenkten Laserstrahlen werden von drei in 50 m entfernt stehenden Kameras aufgenommen. Die Bilder werden nach Schwerpunkt, Intensitätskurve und evtl. Verformung der Strahlen ausgewertet. Diesen Vorgang wiederholt man unter verschiedenen Temperaturen.

Gesteuert wird der gesamte Messvorgang von einem PC, der die aktuelle Temperatur misst, diese regelt, den Laser und die Kameras ansteuert sowie die aufgenommenen Bilder auswertet (vgl. Abb. 2). Zur Zeitersparnis soll ebenfalls das Durchlaufen eines vorgegebenen Temperaturverlaufs mit automatischen Messungen möglich sein.

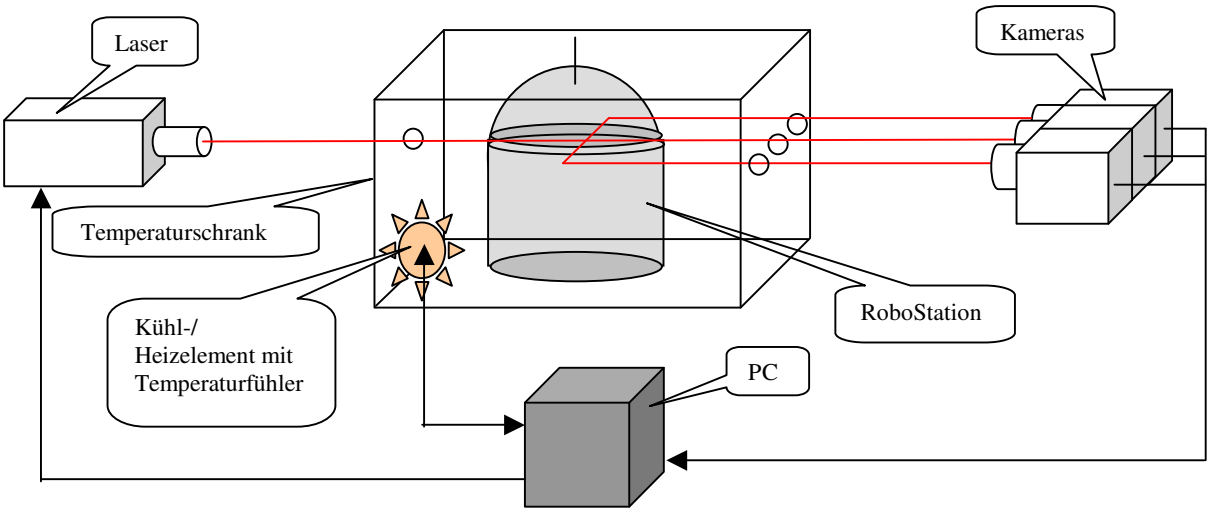

Abb. 2: Prinzipieller Messaufbau

Um die Auswirkungen einer Temperaturänderung festzustellen, vergleicht man die ermittelten Daten hinsichtlich Mittelpunktsverschiebung, Intensitätsänderung, Verformung des Laserstrahls und Temperaturunterschied. Durch Kenntnis der Verzerrung in Abhängigkeit vom Temperaturunterschied lassen sich nun Rückschlüsse auf die Veränderung des Testobjekts ziehen und eine Funktion zum Ausgleich der temperaturbedingten Schwankungen berechnen.

3. Aufbau

3.1 Hardware

Dieses Kapitel beschreibt die zur Kommunikation und Messung mit dem PC notwendigen Bauteile.

Zur Regelung des Peltierelements dient ein RaPID grado-Regler 913 der Firma Hengstler GmbH mit digitaler Schnittstelle RS485 zur externen Sollwertangabe. Des weiteren werden ein Solid State Relais und ein PT 100 Temperatursensor benötigt. Mit Hilfe der im *R*esponse *a*ssisted *PID* Regelverfahren (kurz: RaPID) verwendeten Fuzzy Logic ist es dem Regler wesentlich schneller möglich auf Lastwechsel zu reagieren. Überschwinger werden dabei weitgehend reduziert (siehe Abb. 3).

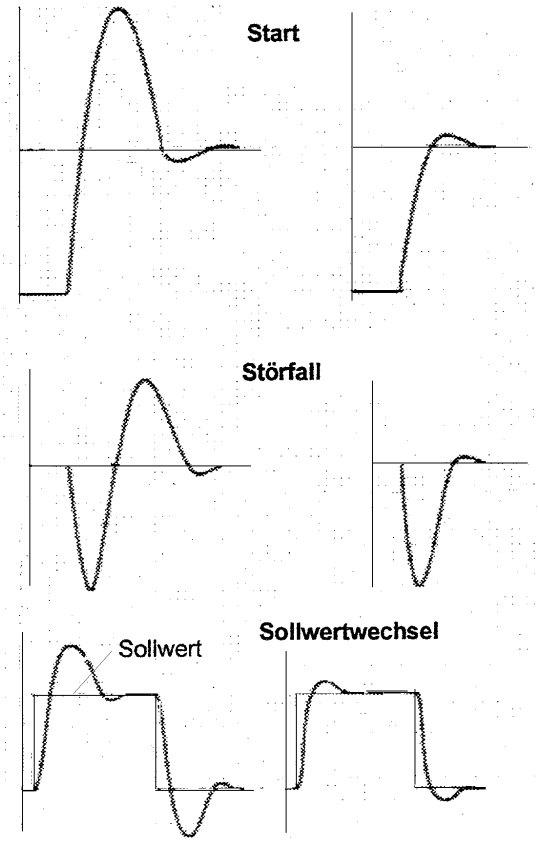

Abb. 3: Vergleich PID-Verhalten (links) zu RaPID-Verhalten (rechts)

Um den Regler mit dem Steuerrechner zu verbinden, braucht man zusätzlich einen RS485 zu RS232 Adapter.
Hierfür eignet sich beispielsweise der SCIA 485 der Firma Hengstler, der speziell für die grado/West-Serie entworfen wurde. Der Adapter sorgt automatisch für Pegelanpassung, Treiberumschaltung und Richtungsteuerung der RS485 Schnittstelle. Der Signalfluss wird mit Hilfe von LEDs angezeigt. Rot bedeutet Signalfluss zum und grün vom Regler. Gelb signalisiert den Idle-Zustand. Das Interface wird über ein internes Netzteil mit 230 V

Wechselspannung versorgt und hat eine Leistungsaufnahme von maximal drei Watt. Er arbeitet ohne Hardware-Handshake. Seine Betriebsbereitschaft wird über den internen Pegelwandler durch das Data System Ready-Pin (DSR) signalisiert.
In der vorliegenden Arbeit wurde jedoch ein HM-485 RS232 - RS485 Schnittstellenumsetzer der Firma Connectivity Expert verwendet werden. Die Richtungssteuerung muss hierbei explizit vom Steuercomputer vorgenommen werden (vgl. Kapitel 3.2). Ansonsten entspricht die Funktionsweise dem SCIA495.

Zur Vergleichstemperaturmessung dient ein Temperaturmesssystem für DALLAS 1820 Temperaturfühler der Firma Hygrotec Messtechnik. Das Gerät hat eine Auflösung von 0,02 °C mit einer Genauigkeit von 0,5 °C und kann bis zu 16 Sensoren verwalten - wovon vorerst drei genügen.

Um die Bilder zu digitalisieren wird ein pciGrabber-4plus der Firma PHYTEC Messtechnik GmbH in Verbindung mit einer Schwarz-Weiß PAL-Kamera der Firma Monacor verwendet.

3.2 RS232- vs. RS485-Standard

Während der RS232-Standard als serieller Com-Port sehr bekannt wurde, ist die RS485-Schnittstelle als vornehmlich in der Industrie eingesetzter Standard (nach EIA-RS485, ISO 8482) privat weniger geläufig.

Die RS232-Schnittstelle benötigt für jede Übertragungsrichtung je eine Leitung. Sie wurde zur lokalen Punkt-zu-Punkt-Kommunikation mit einem Transmitter und einem Receiver entworfen. (Die Kabellänge sollte 30 - 60 m nicht überschreiten.) Die Signale werden durch sog. *unbalanced data transmission*, d.h. durch einen Spannungspegel, der mit einer Referenzspannung (meist Erdung) verglichen wird, repräsentiert. Die maximale Datenrate liegt üblicherweise bei 115200 bps.

Das RS485-Interface dagegen ist für höhere Datenraten (2,5 MB/s) und wesentlich weitere Distanzen (bis 1200 m Kabellänge) ausgelegt. Die Multipoint-Kommunikation unterstützt bis zu 32 Transmitter / Receiver und erfolgt zu meist in einer Master/Slave-Architektur mit Polling - wobei, dem Ethernet vergleichbar, mehrer Quellen an den Bus angeschlossen sein können. Eine Kommunikation wird dabei immer vom Master gestartet und nur von demjenigen Slave mit der passenden Adresse beantwortet. Da die Kommunikation hier über *balanced data (= differential voltage) transmission* im Halb-Duplex-Betrieb erfolgt, benötigt jedes Kabel eine *Twisted Pair* Leitung mit Abschlusswiderstand an jeder Seite (siehe Abb. 4). Man unterscheidet dabei zwei Ausführungen von RS485-Schnittstellen: *Single* und *Double Twisted Pair*. Um für Systeme, die ursprünglich für RS232 entworfen wurden, ebenfalls Multipoint-Kommunikation bereitzustellen, sendet der Slave bei einer Schnittstelle vom zweiten Typ im Gegensatz zu Typ eins auf einer eigenen Leitung.

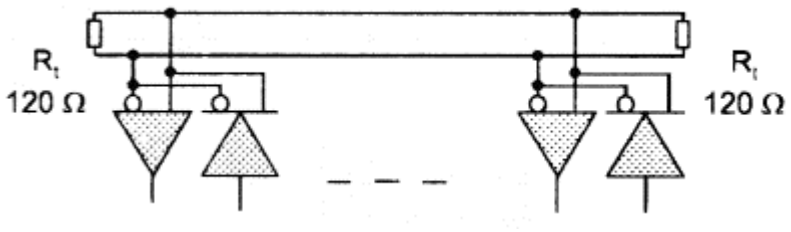

Abb. 4: Prinzipieller Aufbau einer RS485-Schnittstelle

Die sog. *balanced data transmission* basiert auf der Differenz der Signalpegel der beiden Leitungen eines *Twisted Pair* Kabels (siehe Abb. 5). Da im Verlauf der gesamten Übertragungsstrecke auf beide Leitungen die gleichen Störungen wirken, heben sich Störrauschen etc. bei der Differenzbildung praktisch auf. Auf diese Art erreicht man auf Kosten komplexerer - und somit teurerer - Schaltungen sehr hohe Datenraten (> 10 Mbit/s) bei Kabellängen bis 1000 m.

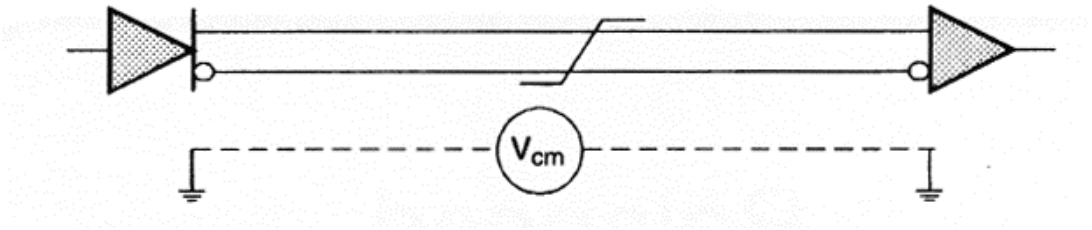

Abb. 5: Prinzipieller Aufbau einer *differential voltage transmission*

3.3 Kalibrierung des CCD-Chips

Da die genauen Daten der Kamera (wie Pixelform, -größe und -abstand des CCD-Chips etc.) nicht bekannt sind, müssen diese zwecks Kalibrierung der Messergebnisse experimentell bestimmt werden.

Ziel der Kalibrierung ist die Umrechnung der Schwerpunktbestimmung von Pixel in Mikrometer. Dazu muss ein Zusammenhang zwischen Verschiebung des Auftreffpunktes des Lasers zur Schwerpunktverschiebung des Bildes hergestellt werden. Hierzu genügt die in Abb. 6 dargestellte Versuchsanordnung.

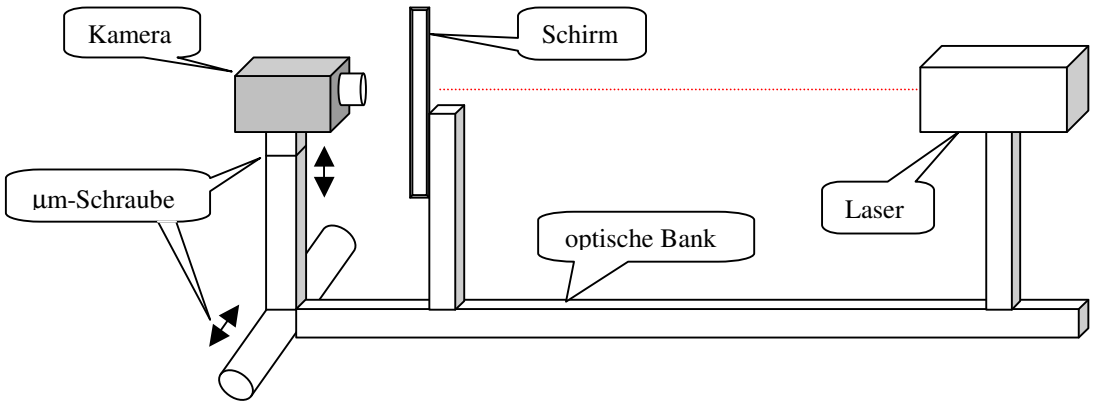

Abb. 6: Aufbau zur Kamerakalibrierung

Zuerst muss die Kamera auf den Schirm scharfgestellt werden, Laser und Schirm sind fest auf einer optischen Bank montiert, so dass die Positionen untereinander sich nicht verschieben können. Die genaue Entfernung des Lasers vom Schirm spielt keine Rolle, er sollte jedoch einen möglichst runden Strahl liefern. In der Realität treten durch den Einsatz von Kameralinsen jedoch Verzerrungen auf, die entsprechend berücksichtigt werden müssen. In der vorliegenden Arbeit wurde daher aus Gründen der Fehlervermeidung auf die Linse verzichtet. Weiterhin wurde der Laserstrahl mit Hilfe einer an Stelle des Schirms eingesetzten Linse direkt auf das CCD fokussiert. Die Kamera wird nun in je eine Richtung um einen festen Wert verschoben. Dabei wird je ein Bild aufgenommen und wie in Kapitel vier beschrieben mit Differenzbildverfahren ausgewertet.

Im optimalen Fall ließe sich ein linearer Zusammenhang zwischen Positionsänderung der Kamera und Schwerpunktsverschiebung feststellen, d.h. bei einer vorgegebenen Genauigkeit von 10 µm und bei einer Verschiebung der Kamera um 10 µm, sollte sich der Schwerpunkt um den gleichen Wert in die gleiche Richtung verlagern.

3.4 Mechanischer Aufbau

Beim mechanischen Aufbau des Temperaturschranks ist v.a. auf thermische Isolation und Erschütterungsfreiheit zu achten.

Um den Aufbau möglichst stabil zu gestalten, wurde zunächst ein quaderförmiger Rahmen 80 cm x 59 cm x 46,5 cm aus Aluminiumprofilen und Winkelstücken erstellt. Der Quader steht auf 8 cm hohen Verlängerungen der Aluprofile mit zusätzlich angeschraubten Füßen. Die benutzten Füße können dabei in jede Richtung gekippt werden um evtl. Unebenheiten auszugleichen. Der fertige Rahmen wurde nun von innen mit Styrodurplatten ausgekleidet. Danach ist im Innenraum keinerlei Kontakt mehr zum Aluminium möglich, welches, im Gegensatz zu Styrodur, ein ausgezeichneter Wärmeleiter wäre. Die Wände messen 60 cm x 42,5 cm und 50 cm x 42,5 cm, Deckel und Boden 68 cm x 49,5 cm. Die Übergänge der Seitenteile werden zusätzlich mit 6 mm breitem Fenster- und Türmoll aus Schaumstoff abgedichtet. Das Oberteil liegt dabei einfach auf den mit Magnethaltern befestigten Seitenteilen auf, so dass es leicht abnehmbar ist. Das Unterteil liegt auf vier an der Unterseite der kurzen Profile festgeschraubten Winkeln auf.

Bei der Wahl der Heizquelle, entschieden wir uns für ein Peltierelement, da dieses durch Umpolung zwischen heizen und kühlen umgeschaltet werden kann. Dieses wird etwa auf halber Höhe an der Innenseite des Behälters hinter einer eingezogen Trennwand befestigt. Die Trennwand besteht ebenfalls aus Styrodur, ist aber nicht durchgängig, sondern schirmt das Peltierelement ab um zu verhindern, dass sich das Messobjekt durch eine direkte Bestrahlung ungleichmäßig erwärmt. Zum gleichen Zweck werden auf den Peltierelement sowie an weiteren Stellen im Raum Ventilatoren angebracht. Dieses Prinzip ist vergleichbar einem Umluftofen, bei dem an jeder Stelle im Wesentlichen die gleiche Temperatur herrscht.

Der mit dem Temperaturregler verbundene Sensor wird im Luftstrom vor dem Peltierelement montiert. Um festzustellen, ob der Temperaturschrank gleichmäßig beheizt ist werden zusätzliche Temperatursensoren in der Nähe des Messobjekts und unter dem Peltierelement angebracht.

Die für den Laser benötigten Öffnungen werden doppeltverglast, um den Temperaturaustausch mit der Umgebung möglichst gering zu halten und gleichzeitig ein Anlaufen der Scheiben zu verhindern. Da der Laserstrahl durch Unreinheiten im Glas nicht unnötig abgelenkt werden soll, sind spezielle planoptische Gläser nötig. Ihr Durchmesser beträgt ein Zoll. Je zwei dieser Gläser werden in ein Plasikrohr eingepasst, dessen Länge genau der Breite des Styrodur entspricht. Diese Einheiten werden in Höhe des zu vermessenden Prismenverbandes wie folgt eingesetzt: ein Doppelglas exakt gegenüberliegend für den direkten Strahl und die anderen um je 5,5 cm versetzt links und rechts daneben. Für ein zusätzliches Beobachtungsfenster, das nach außen ebenfalls mit Styrodur isoliert sein muss, genügt jedoch normales Glas.

Der Laser und die im Temperaturschrank befindliche Strahlteileroptik sind mechanisch miteinander verbunden, wobei sich der Laser vor dem Temperaturschrank optimalerweise auf einem kleinen Tisch mit drei Freiheitsgraden befindet. Die gesamte Apparatur steht auf einem möglichst schweren, stabilen Tisch mit einer geringen Eigenschwingung. Die drei Kameras stehen in etwa 50 m Abstand in gerader Line zu je einem Austrittsfenster dahinter.

3.5 Elektrischer Aufbau

Dieser Abschnitt beschreibt den elektrischen Aufbau der Versuchsanordnung, d.h. vor allem die Anschlüsse an den PC sowie die Verkabelung der Sensoren, Ventilatoren, des Lasers und des Peltierelements innerhalb des Temperaturschranks.

Der Grado913-Regler wird über die Klemmen 13 und 14 mit 230V direkt aus dem Stromnetz versorgt und an den Klemmen 16 (+) und 17 (-) über den RS232 ↔ RS485 Konverter an den Com2-Port des PCs angeschlossen (Klemme 16 mit Klemme drei des Konverters und Klemme 17 mit den Klemmen vier des Konverters).
Den PT100 Temperatursensor schließt man an die Klemmen eins und zwei mit einer zusätzlichen Brücke von zwei nach drei um störende Spannungsanteile, die sich evtl. durch ein langes Kabel ergeben, auszugleichen.
Die Anschlüsse des Reglers sind in Abb. 7 dargestellt.

Der HM-485 Konverter wird hierbei im Zwei-Draht Halb-Duplex Modus betrieben, d.h. der Plus- / Minuspol des Senders wird mit dem Plus- / Minuspol des Empfängers verbunden (Klemme 17 mit Klemme zwei und drei des Konverters und Klemme 16 mit den Klemmen eins und vier des Konverters). Die Schalterstellungen sind T-RTS, R-$\overline{\text{RTS}}$ und DCE.

Der Temperaturdatalogger ist an den seriellen Com1-Port des Steuerrechners angeschlossen und wird über ein Steckernetzteil mit 9 V versorgt. Abb. 8 zeigt die verschiedenen Anschlüsse des Messsystems.

Der Dallas Temperatursensor wird an den Pins drei bis fünf des Temperaturmesssystems verbunden (Abb. 8).

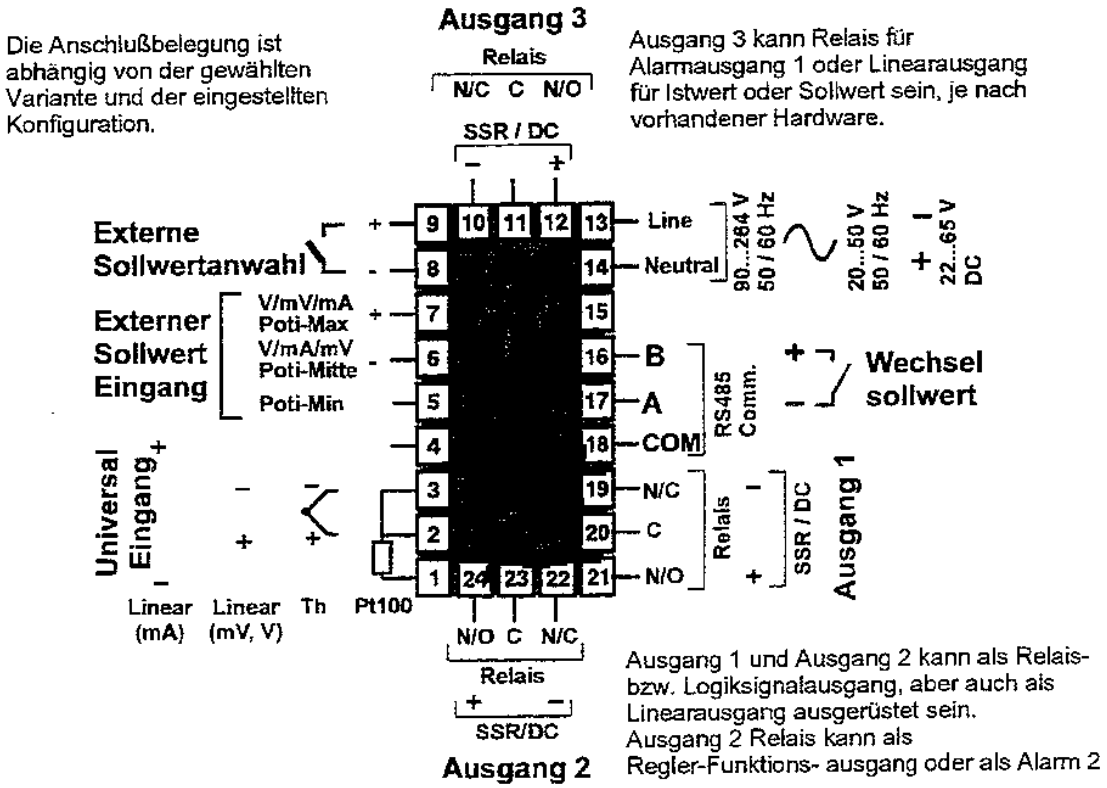

Abb. 7: Anschlüsse grado-Regler 913

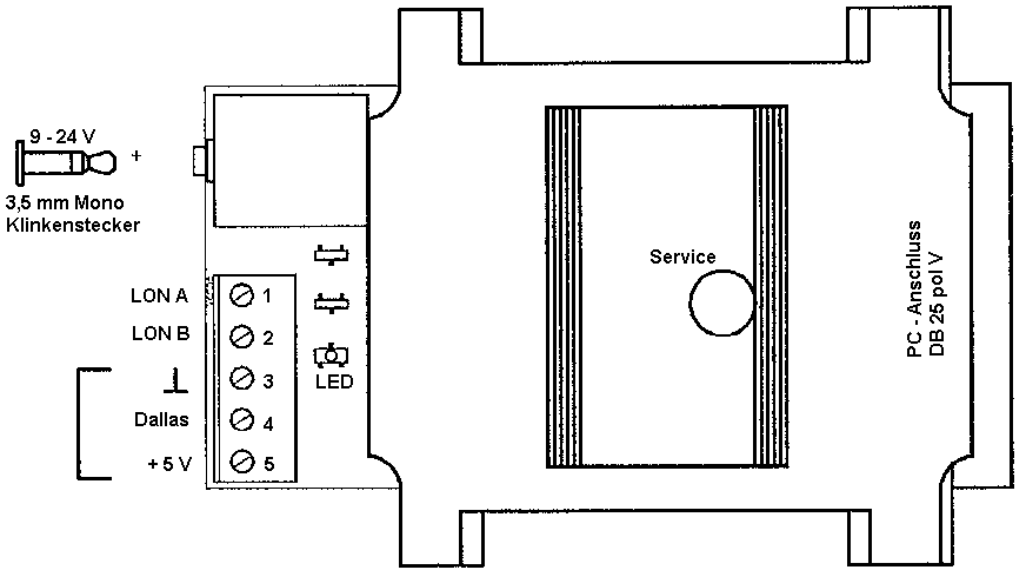

Abb. 8: Anschlüsse des Temperaturmessystems

4. Software

4.1 Kontrollsoftware aus Sicht des Users

Im Folgenden wird die Bedienung der Software beschrieben, die den gesamten Messprozess überwacht und kontrolliert. Auf die Auswertung der gewonnen Informationen wird erst in einem späteren Kapitel eingegangen.

Abb. 9: Kontrollpult der Messstation

Als Benutzer hat man über ein Kontrollpult mit Grundfunktionen Einfluss auf den Messvorgang. Es werden somit keinerlei Kenntnisse über die verwendeten Bauteile und ihre genaue Funktionsweise benötigt. Grundsätzlich hat der User die Wahl zwischen manueller Messung, automatischer Messung und einem Livebildmodus (siehe Abb. 9).

Während es ihm bei der ersten Variante zu jeder Zeit selbst möglich ist die Solltemperatur vorzugeben und einzelne Kameras an bzw. ab zu schalten, eignet sich die automatische Messung für längere Messreihen. Das Programm ist in diesem Modus in der Lage, ein festes Messprofil aus Textdateien einzulesen und selbstständig abzuarbeiten.

Die verwendeten Eingabedateien haben die Endung .txt und kennen drei Zeilentypen, die nach festen Schemata aufgebaut sind (siehe Bsp. 1):

Typ 1: Eine Zeile mit der folgenden Syntax enthält ein gültiges Profiltupel: "%i s, %f°\n". Hierbei wird der erste Wert als Erwärmungsdauer in Sekunden und der zweite als Grad Celsius Angabe erkannt. Die in der linken Spalte angegebene Zeit beginnt erst nachdem die rechts angegebene Temperatur in der gesamten Temperaturkammer erreicht wurde.
Typ 2: Mit ‚#' beginnende Zeilen werden als Kommentare ignoriert.
Typ 3: Zeilen die nicht dem obigen Schema entsprechen, gelten als ungültig und werden ebenfalls ignoriert, jedoch wird eine entsprechende Fehlermeldung in das Messprotokoll geschrieben.

```
60 s                    10°
30 s                    20°
# hier ist Platz für Kommentare
35 s                    30°
100 s                    0°
diese Zeile ist ungültig
```

Bsp. 1: Beispiel für ein Messprofil

Wichtige Bildparameter wie Helligkeit, Kontrast, Farbton und Sättigung können jederzeit vom Benutzer geändert werden. Dies geschieht über ein grafisches Kontrollpult (siehe Abb. 10), das für jede Kamera getrennt sowohl das aktuelle Bild und die aktuellen Einstellungen (links), sowie die Auswertung der Daten (rechts) anzeigt.
Da die Änderungen erst bei der nächsten Aufnahme wirksam werden, eignet sich der Livebildmodus hierzu besonders, weil man Änderungen direkt mitverfolgen kann.

Die Auswertung eines neu gegrabbten Bildes erfolgt automatisch. Die Ergebnisse werden automatisch sowohl im Kontrollpult der jeweiligen Kamera ausgegeben (siehe Abb. 10), als auch in einer Textdatei mit aktuellem Datum im Ordner Results gespeichert (vgl. Bsp. 3). Der Dateiname setzt sich aus dem Präfix „Results" bei manuellen bzw. „AutoResults" bei automatischen Messungen und dem aktuellen Datum (getrennt durch Unterstich) zusammen. Für Messungen, die am 18. November 2001 aufgenommen wurden, wäre das *c:\Verena\Results\Results_18_11_2001.txt bzw. AutoResults_18_11_2001.txt.* Die Ausnahme hierbei bildet der Livebildmodus bei dem keine Ausgabe in die Datei erfolgt. Der Unterschied im Ausgabentext besteht darin, dass bei einer manuellen Messung die Angabe der Heizdauer fehlt und im Textfenster des Kontrollpults zusätzlich einige statistische Daten ausgegeben werden.

```
Aussteuermaximum: 253
Übersteuerung bei 0 Pixel.
Profilname: C:\Verena\profiltest.txt
ID: 1
Temperatur: 25.9 °
Dauer: 400 s
Schwerpunkt: (327.565, 237.974)
Mittelwert: 105
Kamera: 1
```

Bsp. 2: Ausgabetext einer automatischen Messung

Profilname: C:\Verena\profiltest.txt
ID: 1
Temperatur: 10.0 °
Dauer: 100 s
Schwerpunkt: (479.014, 239.934)
Mittelwert: 152
Kamera: 1

Bsp. 3: Beispiel für Ergebnisdatei einer automatischen Messung

Danach werden sowohl Laser als auch Kameras und Peltierelement eigenständig abgeschaltet. Der Wechsel zwischen den beiden Betriebsarten ist jederzeit durch den Knopf „Messabbruch“ möglich.

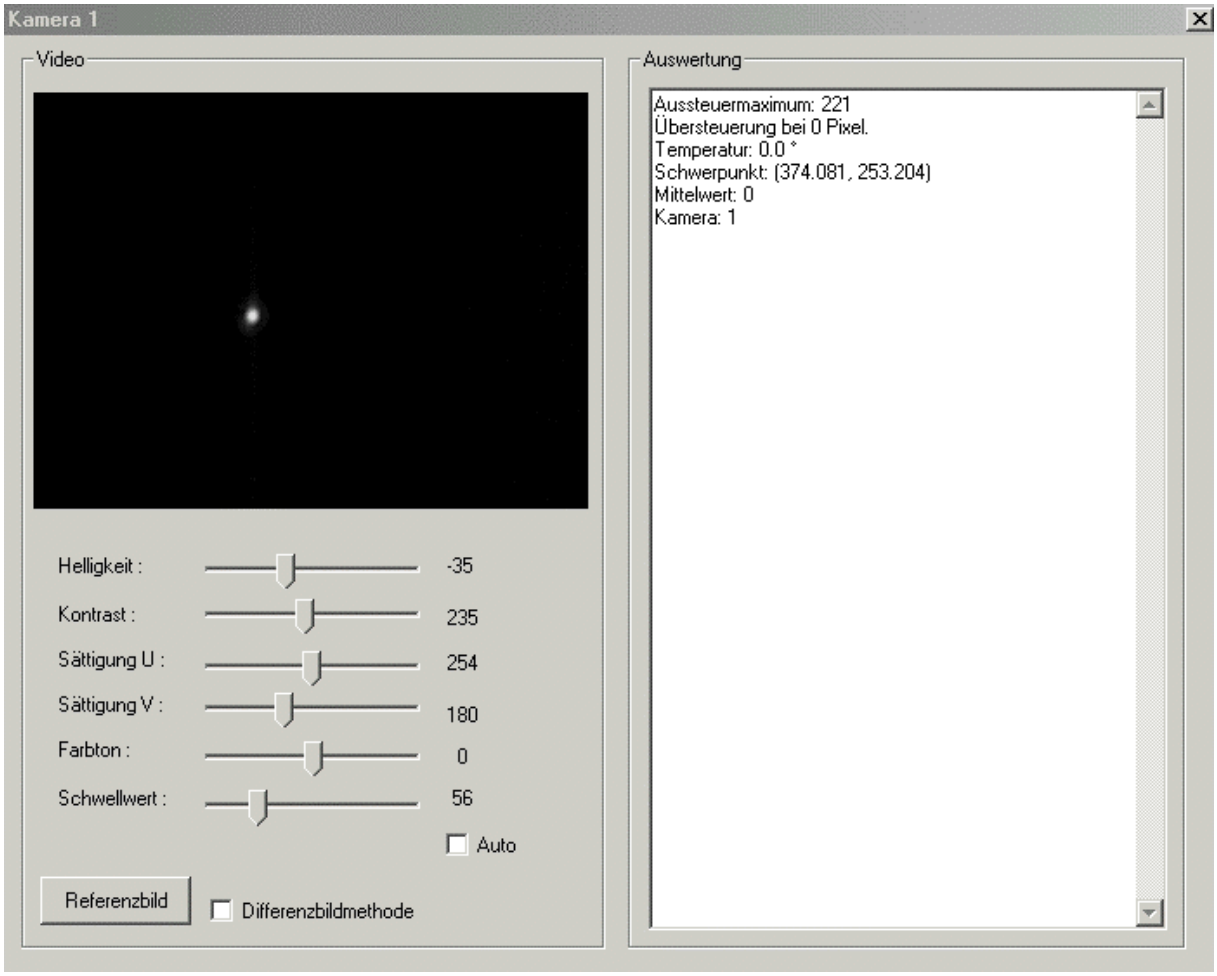

Abb. 10: Kontrollpult einer Kamera

Der Knopf „Referenzbild“ im Kontrollpult einer Kamera dient dazu ein Referenzbild aufzunehmen. Wenn das Kontrollkästchen „Differenzbildmethode“ vor einer Messung aktiviert wurde, wird ein Differenzbild zwischen dem aktuell aufgenommenen und dem Referenzbild zur Auswertung benutzt um so den Hintergrund auszublenden.

Des weiteren hat der Benutzer durch Setzen des „Auto“-Flags die Wahl zwischen einem automatisch berechneten Schwellwert (in diesem Fall der Mittelwert über alle Helligkeitswerte) oder einem mit Hilfe des Schwellwertsliders selbst angegebenen Schwellwert.

4.2 Kontrollsoftware aus interner Sicht

In diesem Unterkapitel wird auf die wichtigsten Grundzüge der Programmierung der Steuersoftware und die Einstellungen der Anlage eingegangen.

Auf die Steuerung von Laser und Kameras wird dabei nicht näher eingegangen, da sich diese aus interner Sicht besonders einfach gestaltet, da ihr Ein- und Ausschaltsignal sowie sämtlich Bildparameter der Kamera vom Framegrabber gesteuert werden.

4.2.1 Die Klassenhierarchie

Aufgrund der Vorteile wie Übersicht und Wiederverwendung ist die Software für jedes Fenster und jedes Gerät in einer eigenen Klasse gekapselt. Diese folgen einer strengen Hierarchie, die sich nicht aus Vererbung ergibt, sondern dadurch, dass eine höherliegende Klasse eine Instanz der direkt darrunterliegenden als Variable hat. In Abb. 11 wird diese Abhängigkeit durch Pfeile verdeutlicht.

Die verschiedenen Aufgaben der einzelnen Klassen sind in Tabelle 1 aufgelistet. Die Tatsache, dass sowohl *KontrollPult* als auch *KameraKontrolle* eine Instanz desselben Grabbers haben, ist kein Problem, da *klimaControl* diese nur zur anfänglichen Initialisierung des Geräts sowie zur Steuerung des Lasers nutzt, während *KameraKontrolle* die Veränderung der Bildparameter Helligkeit, Kontrast etc. und das Grabben an sich übernimmt. Da jede Änderung der Parameter erst beim Digitalisieren des nächsten Bildes wirksam wird, sind die Einstellungen innerhalb eines Bildes, selbst bei überlappenden Befehlen, immer konsistent.

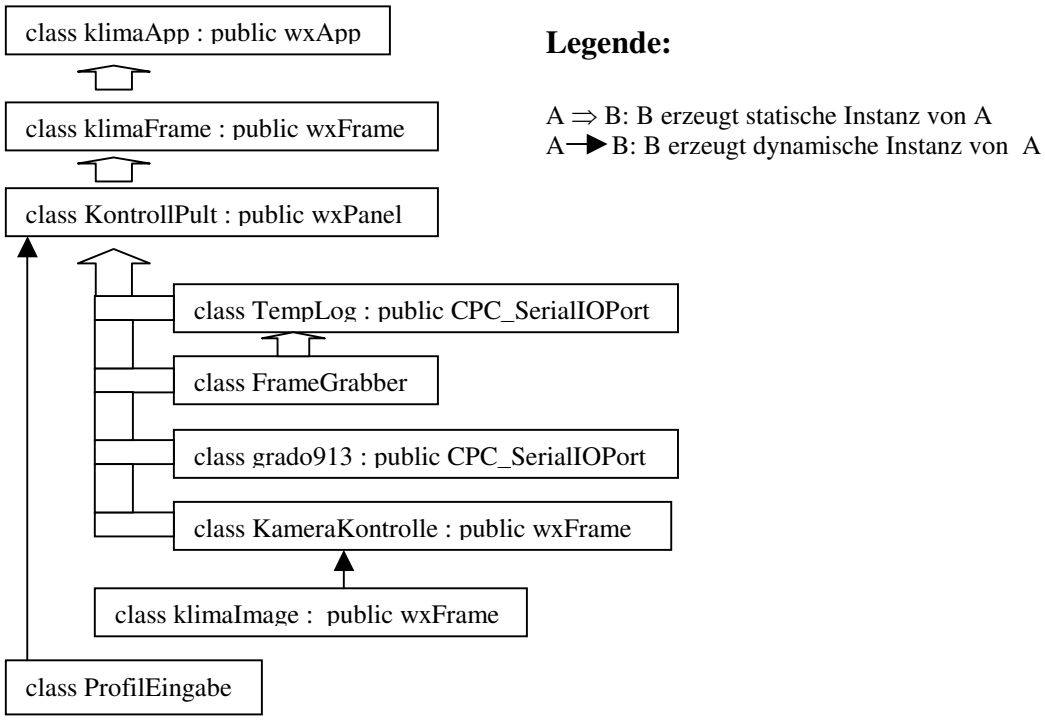

Abb. 11: Klassenhierarchie

klima_app:	Hauptprogramm, öffnet Hauptfenster und Menüleiste
klima_Frame:	Fenster des Hauptkontrollpults, öffnet Hauptkontrollpult
KontrollPult:	stellt Hauptkontrollpult dar, startet bei Useranfrage passende Routine kontrolliert Regler, Datalogger und Kamerakontrollpult, initialisiert Grabber
KameraKontrolle:	stellt Kamerakontrollpult dar, startet bei Useranfrage passende Routine
FrameGrabber:	stellt Grabberroutinen aus Bibliothek bereit
grado913:	Kommunikation mit dem Regler, Lesen und Ändern von Parametern
TempLog:	empfängt Daten des Datalogger und stellt sie bereit
klimaImage:	Bildauswertung
ProfilEingabe:	Einlesen und bereitstellen eines Messprofils aus einer Textdatei

Tabelle 1: Aufgaben der verschiedenen Klassen

4.2.2 Auslösen eines Messvorgangs

Die für automatisches und manuelles Messen bzw. für den Live-Modus benötigten Steuerprozeduren gehören zu der Klasse *KameraKontrolle* (dazu vergleiche Klassenhierarchie in Abb. 10). Es werden nur zwei Prozeduren benötigt, da der Live-Modus die Funktion für manuelles Messen mitbenutzt. Der Hauptunterschied zwischen den Messroutine für manuelles und automatisches Messen liegt in der Ausgabe der Auswertungsdaten, da bei einer automatischen Messung noch die Zeit zwischen dem Erreichen der Solltemperatur und dem Messen berücksichtigt wird und im Live-Modus nur eine Ausgabe auf den Bildschirm erfolgt.

Die Klasse *KameraKontrolle* enthält das gewünschte Messprofil als Membervarible. Das Anfordern eines Messvorgangs wird im automatischen Modus nicht vom User, sondern von der Prozedur *KontrollPult::OnIdle(wxCommandEvent&)* übernommen und arbeitet die Anforderung sukzessive ab, solange die globale Variable *MessenAuto* auf *cTRUE* steht. Dieselbe Prozedur ist für das ständige Auslösen einer Messung im Live-Modus zuständig. Doch während bei der Bearbeitung einer automatischen Messung zusätzlich manuelle Messung gemacht werden können, schließen sich automatisches Messen und Lifebild aus (Dazu mehr in Kapitel 4.2.3).

Das Auslösen eines manuellen Messvorgangs wird ebenfalls von der Klasse *KontrollPult* übernommen. Sie beinhaltet für jede Kamera einzeln bzw. alle zusammen eine Funktion die auf einen Klick des Benutzers im Hauptkontrollpult (Abb. 9) reagiert und dann die Routine *KameraKontrolle::MessenManuell* mit den passenden Parametern für manuelles Messen startet. Beim Aufruf einer manuellen Messung wird zusätzlich unterschieden, ob es sich um eine exakte Messung oder um ein Livebild handelt. Livebilder sind immer ungemittelt, wogegen bei einer exakten Messung *DefAnzMittelung* Bilder gemittelt werden. Je größer die in *Parameter.h* festgelegte Anzahl der Bilder, desto resistenter ist das Ergebnis gegen Störrauschen.

Falls bei der Messung ein Fehler aufgetreten ist und somit die Messfunktion den Wert *cFalse* zurückliefert, wird eine Fehlermeldung an den Benutzer ausgegeben.

4.2.3 Heizen und Messen in der Idle-Loop

Die zentrale Schaltroutine des Programms ist die im Hintergrund laufende Idle-Loop, die, sobald sich die CPU im Leerlauf befindet, für die Aktualisierung des Temperaturistwerts im Kontrollpult (vgl. Abb. 9), die Aufnahme eines Livebildes und das Abarbeiten eines Messprofils sorgt.

Da die Kommunikation mit dem Regler relativ viel Zeit beansprucht, wird die aktuelle Temperatur nicht jedes Mal ausgegeben. Die Aktualisierungsrate *RateIstwertAktualisierung* kann man in *Parameter.h* einstellen. Die Ausgabe erfolgt auf eine Nachkommastelle genau in °C.

Um ein Livebild zu darzustellen, überprüft die Routine jedes Mal die „Livebild"-Flags (siehe Abb. 9) und löst, wenn ein Flag gesetzt ist, den entsprechenden Messvorgang aus. Die Bilder sind ungemittelt und ihre Auswertung wird nicht in die Datei protokolliert.

Die Temperaturvorgabe erfolgt durch die Prozedur *KontrollPult::SetTemp*, die sowohl von der Idle-Loop, als auch von *KontrollPult::OnApply* genutzt wird. Hierbei sorgt zweitere für das Setzen eines neuen manuell angegebenen Sollwert (bei Bestätigung durch den "OK"-Knopf in Abb. 9) und erstere für das Setzen des als nächstes zu bearbeitenden Sollwerts im Profil. Des weiteren setzt *KontrollPult::SetTemp* die Fließkommavariable *temp* in das spezielle Format des Reglers, d.h. einen String aus vier Digits plus Vorzeichen, um und ruft damit die Reglerfunktion *grado913::SetParameter* auf, um den Sollwert zu aktualisieren. Als letztes setzt sie die globale Variable *TempKonstErreicht* um der Idle-Loop zu signalisieren, dass die Solltemperatur erreicht wurde.

Das Heizen bzw. Kühlen an sich übernimmt der grado-Regler durch Steuerung des Peltierelements. Die Idle-Routine überwacht lediglich den Fortschritt und verhindert eine automatische Messung solange nicht eine gleichmäßige Temperaturverteilung im gesamten Temperaturschrank erreicht wurde. Um dies zu gewährleisten, nutzt die Funktion die drei Temperatursensoren des Dataloggers als Vergleichswerte. Der Programmierer kann in *Parameter.h* einen Toleranzwert *TempFehler* festsetzen und so das Zielintervall an seine Genauigkeitsbedürfnisse anpassen. So lange eine Temperaturregelung läuft, überprüft die Idle-Routine jedes Mal die Temperaturwerte der Dallassensoren und des PT100 bis diese auf einen Fehler von *TempFehler* Grad übereinstimmen. Die im Profil angegebene Zeitspanne beginnt erst, wenn diese Übereinstimmung über eine in *Parameter.h* gewählte Anzahl von Durchläufen *AnzKonstPeriode* stabil bleibt. Wird die Temperaturgleichheit nur kurzzeitig erreicht, bsp. bei einem Überschwinger, beginnt das Zählen der Intervalle von vorne.

Startet der User ein zuvor ausgewähltes Messprofil durch betätigen des „Start"-Knopfes, werden alle für die Idle-Loop nötigen Werte gesetzt, so dass diese bei ihrem nächsten Aufruf mit der Abarbeitung beginnen kann. Hierbei signalisiert die globale Variable *AutoMessung*, ob ein Messprofil abzuarbeiten ist, *ProfilCount* gibt die Nummer des aktuell zu bearbeitenden Tupels an und *goal* speichert den Zeitpunkt der nächsten Messung. Des weiteren gibt *KontrollPult::OnStart* sogleich den ersten Auftrag an die Heizroutine und setzt gleichzeitig die Variable *TempKonstErreicht* zurück auf *cFALSE*.

Ist der Temperaturschrank gleichmäßig temperiert und die vorgegebene Wartezeit abgelaufen, schaltet die Idle-Loop den Laser ein mit der Funktion *KontrollPult::laser_on*), initiiert eine

automatische Messung mit jeder Kamera durch die Prozedur *KontrollPult::MessenAuto* und schaltet den Laser mit der Funktion *KontrollPult::laser_off* wieder ab. Sind alle (Zeit, Temperatur)-Tupel abgearbeitet, informiert die Routine den Benutzer über das Ende des Messvorgangs.

4.2.4 Die Steuerung des Framegrabber

Vor dem ersten Zugriff muss jede Kamera einzeln mit dem Aufruf *Init* initialisiert werden. Der Treiber reserviert hierbei automatisch den benötigten Speicherplatz für ein Bild pro Kamera.
Diese und alle anderen Funktionen des Framegrabber werden durch die Bibliothek *Gr4cdll.dll* zur Verfügung gestellt und durch *Frame_Grabber.h* bzw. *Frame_Grabber.cpp* gekapselt.

Der Grabber kann ein Vollbild sowohl einzeln als zwei Halbbilder als auch interlaced, d.h. verzahnt aufnehmen. Des weiteren hat der Programmierer die Wahl zwischen Momentaufnahme und ständiger Aktualisierung eines Bildes. Diese und andere grundsätzliche Eigenschaften wie Bildgröße, Farbformat und -system werden am Anfang mit der Funktion

SetImage(Kameranummer, PositionHorizontal1, PositionVertikal1, Breite1, Länge1, PixelproZeile1, Zeilenanzahl1, Farbformat1, PositionHorizontal2, PositionVertikal2, Breite2, Länge2, PixelproZeile2, Zeilenanzahl2, Farbformat2, Interlaced, Schnappschuss)

automatisch eingestellt, wobei 1 für ungerade und 2 für gerade Halbbilder steht. Weitere Parameter wie Helligkeit, Kontrast, Farbton und Sättigung können einzeln über Set- bzw. Get-Funktionen gesetzt und ausgelesen werden.

Da für die angestrebte Auswertung eine Momentaufnahme der jeweiligen Laserstrahlabweichung genügt, bei der lediglich die Helligkeit der einzelnen Pixel und nicht ihre Farbe interessiert, wird ein Schnappschuss in Schwarz-Weiß gewählt. Die Halbbilder werden der einfachen Auswertung halber gleich richtig verzahnt und die Einstellungen für ungerade und gerade Bilder sollen gleich sein. Für die beste Ausnutzung der Ressourcen, werden die allgemein benötigten Parameter in *GrabberParameter.h* wie folgt gesetzt (siehe Listing 1):

```
#include "KameraSW.h"   // oder: #include "KameraCol.h"
#include "GrabberKonstanten.h"

#define DevNo                1
#define channel1             6
#define channel2             7
#define channel3             8
#define Interlaced           1        // 1 = an, 0 = aus
#define SingleShot           1        // 1 = an, 0 = aus

// Grabparameter für gerade Halbbilder wie für ungerade
#define Ehpos                Ohpos
#define Evpos                Ovpos
#define Ehsize               Ohsize
#define Evsize               Ovsize
#define Eppl                 Oppl
#define Elines               Olines
#define EColFormat           OColFormat
```

```
// Bildparameter
#define minHelligkeit          -128
#define maxHelligkeit          127
#define minKontrast            0
#define maxKontrast            511
#define minHue                 -128
#define maxHue                 127
#define maxU                   511
#define minU                   0
#define maxV                   511
#define minV                   0

#define ColorDepth             8
#define ImageSize              (((ppl + 3) & ~3) * lines)
```

Listing 1: Auszug aus *GrabberParameter.h*

Für die kameraspezifischen Werte (siehe Tabelle 2) muss je nach verwendeter Kamera entweder *KameraCol.h* für die NTSC-Farbkamera oder *KameraSW.h* für die PAL-Schwarz-Weiß-Kamera eingebunden werden. An dieser Stelle werden u.a. alle Bildparameter für ein ungerades Halbbild gesetzt. Um immer ein konsistentes Vollbild zu erhalten, werden die entsprechenden Werte für das gerade Halbbild in *GrabberParameter.h* immer gleich den Parametern für das ungerade gesetzt.

Parameter:	KameraSW:
ppl	640
lines	479
DefaultHelligkeit	-35
DefaultKontrast	235
DefaultHue	0
DefaultU	254
DefaultV	180
// Parameter für Set_Color_System	
ColSystem	PAL_BDGHI
// Parameter für Set_AGC	
AGC	0
ChromaAGC	0
adaptiveAGC	0
// Parameter für Set_BW	
BW	1
// Parameter für LumaControl	
Range	1
Core	0
// Parameter für Set_Image	
Ohpos	8
Ovpos	0
Ohsize	ppl
Ovsize	lines/2
Oppl	648
Olines	486/2
OcolFormat	Y8

// Parameter für Schwerpunktberechnung	
Pixel2MikrometerX	1
Pixel2MikrometerY	1
DefaultSchwellwert	0

Tabelle 2: Kameraspezifische Parameter

Um einen Schnappschuss der aktuellen Szene zu erhalten, muss der Grabber gestartet und nach einigen Halbbildern wieder gestoppt werden. Selbst wenn in dieser Zeit mehrere vollständige Halbbilder verstrichen sind, wird nur das jeweils erste gerade bzw. ungerade digitalisiert. Hierbei liefert *GetCounter* die Anzahl der digitalisierten Halbbilder, welche bei erfolgreichem Grabben eines Halbbildes immer zwei sein sollte. Bei fortlaufendem Grabben zählt *GetCounter* die bisher vollständig digitalisierten Bilder, d.h. bei sechs wurde ein gerades und ein ungerades Bild gegrabbt und je zwei mal aktualisiert. *DataPresent* misst den Digitalisierungsfortschritt und gibt Werte von 0 bis 15 zurück. Beides wird zusammen mit *GetError* zur Überprüfung des Grabvorgangs genutzt. *GetError* liefert den letzten aufgetretenen Grabberfehler und unterscheidet dabei zwischen den in Tabelle 3 aufgezählten Fehlercodes:

0: kein Fehler aufgetreten
1: Device nicht gefunden
2: Register nicht vorhanden
3: Initialisierung fehlgeschlagen
4: Grabber nicht gefunden
5: unbekannter Parameterwert
6: nicht unterstützt
7: Update benötigt
8: kein PhytecGrabber gefunden
9: kein Acknowledge
10: ungültige Adresse
11: Schreiben nicht möglich

Tabelle 3: Fehlercodes von *GetError*

Die Digitalisierung eines Schnappschusses, Fehlerüberprüfung und -meldung, sowie die ggf. vorgenommene Mittelung mehrerer nacheinander digitalisierter Bilder zu einem geschieht in der Routine *grab*. Aufrufparameter sind die Adresse unter der die Funktion das Bild als *wxImage* ablegen soll und die Anzahl der zu mittelnden Bilder.

Da die von Phytec mitgelieferten Treiber und Routinen kein fertiges Windows-Bitmap, sondern nur Rohdaten liefern, wird die Umwandlung der zunächst gemittelten Rohdaten in ein Graustufenbild von der Prozedur selbst übernommen. Jedes Datum repräsentiert einen Grauwert. Grau bedeutet, das alle Farbanteile gleich stark vertreten sind. Da das durch ein *wxImage* repräsentierte Bild ein RGB-Bild ist, muss jedem Pixel mit der Funktion *SetRGB* der gleiche Wert zugewiesen werden. Die Adresse des ersten Grauwerts liefert die Funktion *GetAddress*.

Zur Mittelung der einzelnen Bilder wird ein zusätzlicher Bildspeicher *MeanRawImage[ppl][lines]* vom Typ *short* benötigt, der zunächst mit Null initialisiert wird. Hier werden die einzelnen Bildwerte pixelweise aufaddiert und am Ende durch die Anzahl der Bilder geteilt. Erst jetzt kann aus den gemittelten Rohdaten das fertig gegrabbte Bild erstellt werden. Die Anzahl der zu mittelnden Bilder wird als Eingabevariable übergeben.

4.2.5 Temperaturregelung mit dem grado913-Regler

Der Regelvorgang an sich wird komplett vom Grado913 übernommen, so dass der Programmierer, falls gewünscht, nur die Regelparameter anpassen muss.

Um die Kommunikation zwischen Regler und PC zu gewährleisten, muss in *GradoParameter.h* die serielle Übertragung auf folgenden Parametern eingestellt sein (vgl. Listing 3):

```
#define Serial_Grado          Serial_1        // entspricht Com2
#define Regleradresse         1
#define ReceiveDelay          2               // Pause zwischen Senden und Empfangen
#define baud                  b4800           // 4800 Baud
#define stopbit               1               // ein Stopbit
#define parity                evenP           // gerade Parität
#define databits              7               // sieben Datenbits
#define SerialBufferInSize          128       // Puffer für ein- ....
#define SerialBufferOutSize         128       // .....bzw. ausgehende Daten
```

Listing 3: Auszug aus *GradoParameter.h*

Des weiteren muss jede Routine, die mit dem Regler kommuniziert sicherstellen, dass sich der HM-485 Konverter im jeweils richtige Sende-/Empfangsmodus befindet. Da der Adapter dies nicht selbst vornimmt, wird das entsprechende Bit von den Funktionen *CPC_SerialIOPort::WL_SEND* zum Senden an der Regler und *CPC_SerialIOPort::WL_RECV* zum Empfangen der Antwort übernommen. Alle zu sendenden Zeichen müssen en-bloc gesendet werden, da der Regler Zeichen mit zu großen Abstand nicht mehr als zusammengehörigen Befehl interpretiert. Deshalb wird je ein Sende- und ein Empfangspuffer angelegt, die vor die Weiterverarbeitung (senden oder auslesen und verarbeiten der Zeichen) beschrieben wird.

Die Prozedur *Status* prüft, ob der Regler funktionsbereit ist und speichert das Ergebnis in der boolschen public - Variable Reglerstatus. Hierbei steht Eins für aktiv und Null für inaktiv oder nicht vorhanden.

Für das Auslesen und Ändern eines Parameters sind die Funktionen *SetParameter* und *ParameterAbfrage* zuständig, die im Falle eines Erfolges *cTRUE* und sonst *cFALSE* zurückliefern.

Für die Kommunikation mit dem Regler muss ein bestimmtes Datenformat vorliegen. Deswegen muss der data-String von *SetParameter* um ein weiteres Byte ergänzt werden, wobei diese letzte Ziffer sowohl das Vorzeichen, als auch die Position des Dezimalpunkts kodiert (vgl. Tabelle 4). Hierbei muss beachtet werden, dass nicht alle Werte, die sich aus der Konstruktionsvorschrift ergeben, für jeden Parameter gültig sind. So sind für einige Parameter nur positive Zahlen oder nur solche mit nicht mehr als einer Nachkommastelle erlaubt.

String:	temp:
abcd0	+abcd
abcd1	+abc.d
abcd2	+ab.cd
abcd3	+a.bcd
abcd5	- abcd
abcd6	- abc.d

abcd7	- ab.cd
abcd8	- a.bcd

Tabelle 4: Kodierung von Vorzeichen und Nachkommastellen

Die Umrechnung erfolgt in einer eigenen Funktion *grado913::Data2Dez*, die aus den fünf Eingabeparameter vom Typ *unsigned char* den zugehörigen Dezimalwert bestimmt und zurück liefert. Falls keine gültigen Werte übergeben wurden, wird *InvalidValue* als ungültiger Wert zurück gegeben, was einem Wert entspricht, der vom Regler nach obigem Schema nicht erzeugt werden kann (in diesem Fall 0.00001).

Das Setzen eines Parameters durch *grado913::SetParameter* erfolgt in zwei Schritten. Zuerst erfolgt die sog. Übernahmevorbereitung, in der die Software dem Regler die Art des Parameters und die neuen Daten zu Verfügung stellt, diese aber noch nicht ändert. Die tatsächliche Änderung des Parameters wird erst nach dem Übernahmebefehl *grado913::Ubernahme* ausgeführt - vorausgesetzt der erste Befehl liefert *cTRUE* als Zeichen, dass er zur Übernahme bereit ist und die Daten gültig sind.

4.2.6 Temperaturmessung mit Hilfe des Dataloggers

Der für die Regelung wichtige PT100 Sensor ist direkt am Regler angeschlossen und wird auch über diesen abgefragt. Die restlichen Werte werden mit Hilfe des Dataloggers ausgelesen, der alle Nutzdaten blockweise ohne Anfrage des Steuerrechners sendet. Diese Datenblöcke sind immer wie in Bsp. 6 aufgebaut und lassen sich daher leicht automatisch verwalten. Ein Datenblock beginnt immer mit @ und endet mit $. Dazwischen werden bei eins beginnend für jeden Kanal erst eine Zeile mit Konfigurationsdaten und dann eine Zeile mit den Messwerten des Kanals gesendet. Jede Zeile wird mit <CR> abgeschlossen. Die Konfigurationsdatenzeile beginnt mit dem Kennbuchstaben ‚I', gefolgt von der logischen Kanalnummer (zwei Zeichen), Physikalische Fühlerkennung (zwei Zeichen), der Hardware-Kennung (zwei Zeichen), der Seriennummer (12 Zeichen) und der CRC-Prüfsumme (zwei Zeichen). Die Messwerte-Datenzeile beginnt mit dem Kennbuchstaben ‚V', gefolgt von der logischen Kanalnummer (zwei Zeichen), den Messdaten (vier Zeichen) und der CRC-Prüfsumme (acht Zeichen).

```
@<CR>
I010110E0223C000000B1<CR>
V0108DA7D<CR>
I02011050013C00000021<CR>
V0208C276<CR>
I030110B0093C00000017<CR>
V0308CCF9<CR>
$<CR>
```

Bsp. 6: Beispiel für einen empfangenen Datenblock bei drei Temperatursensoren

Die Prozedur *TempLog::ReceiveData* wertet dabei die vom Messmodul geschickten Daten aus und aktualisiert die Temperaturen, die in den *public* Variablen *Sensor1*, *Sensor2*, und *Sensor3* gespeichert sind. Bei erfolgreichem Auslesen der Sensoren liefert sie *cTRUE* zurück.

4.3 Auswertung der Videodaten

Die Auswertung der Videodaten beschränkt sich der Einfachheit halber auf eine Schwerpunktsberechnung der Helligkeitswerte um so die Mitte des Laserstrahls zu bestimmen.

Da die Farbe Weiß durch den Wert 255 und Schwarz durch den Wert 0 - oder gegebenenfalls durch einen Offsetwert nahe Null- repräsentiert wird, lässt sich der Schwerpunkt leicht über zwei gewichtete Summen berechnen. Um das Helligkeitsmaximum in X-Richtung zu bestimmen, addiert man alle Helligkeitswerte abzüglich eines gegebenen Schwellwertes (wobei nur nicht negative Summanden addiert werden dürfen) einer Spalte und gewichtet diese mit der Spaltennummer. Die Teilsummen werden addiert und durch die ungewichtete Summe aller Helligkeitswerte dividiert. Die Y-Koordinate des Schwerpunkts berechnet sich auf die gleiche Weise, nur mit transponierter Helligkeitsmatrix.
Als Schwellwert kann beispielsweise die Grundhelligkeit eines Bildes dienen, die über die Funktion *klimaImage::GetBlackOffset* erreicht werden kann, indem man pixelweise alle Graustufenwerte des Bildes addiert und durch die Pixelanzahl teilt. Alternativ kann über den Regler „Schwellwert“ im Kamerakontrollpult ein beliebiger Schwellwert eingestellte werden, der möglichst über dem Mittelwert liegen sollte.

Falls ein Differenzbild gewünscht wird, muss dieses zuerst durch eine Subtraktion des aktuellen Bildes mit einem gegebenen Referenzbild erzeugt werden. Auch hier muss darauf geachtet werden, dass ein Grauwert nicht kleiner als Null wird, also wird beim Differenzbild entweder nur der Betrag der Differenz betrachtet oder alle Werte unterhalb von Null werden abgeschnitten. (In der vorliegenden Projektarbeit wurde zweitere Methode implementiert.)
Ob nun das Differenzbild oder das aktuelle Bild ausgewertet wird, entscheidet die Auswertungsroutine *KameraKontrolle::Auswertung* indem sie überprüft ob das „Differenzbildmethode“-Flag vom User gesetzt wurde. Diese Entscheidung wird für jede Kamera einzeln getroffen. Als Referenzbild dient das jeweils zuletzt gemachte Bild bevor der „Referenzbild“-Knopf im Kamerakontrollpult gedrückt wurde.

Zusätzlich wird im Auswertungsfenster die maximale Aussteuerung und die Anzahl der übersteuerten Pixel angegeben, d.h. die Anzahl der Pixel mit einem Grauwert gleich 255.

5. Fazit und Ausblick

Ziel der vorliegenden Arbeit über die Konstruktion eines Messstandes zur Untersuchung der thermischen Ausdehnung bei hochpräzisen Optiken am Beispiel der RoboStation der AndroTec GmbH war es, einen erfolgreichen Ausgleich der temperaturbedingten systematischen Abweichungen zu ermöglichen.
Obwohl im Laufe der Arbeit keine komplette Messvorrichtung verwirklicht wurde, stellt sie jedoch einen wichtigen Grundstein dar. So wurde nicht nur das komplette Steuerprogramm, sondern auch mögliche Baupläne entwickelt und auf ihre messtechnische Machbarkeit geprüft. Die Arbeit hat somit ihr Ziel erreicht.
Im Folgenden werden noch einige weiterführende Anregungen vorgestellt.

Zur optimalen Schwingungsdämpfung der Messvorrichtung und somit für möglichst genaue Messergebnisse wird die Verwendung eines Granittischs empfohlen. Ferner gilt es, die Verschiebung der Kameras gegenüber dem Schrank möglichst minimal zu halten, weshalb sich bei den Kameras die gleiche hohe Dämpfung empfiehlt.
Die relative Verschiebung der Kameras untereinander, beispielsweise durch Temperaturänderung, ist ebenfalls nicht zu vernachlässigen. Entscheidend ist hierbei der Ausdehnungskoeffizient des Trägermaterials. Um die Distanz ausreichend stabil zu halten, muss entweder ein Material mit genügend kleinem Ausdehnungskoeffizienten, bsp. eine spezielle Keramik, gewählt werden oder ein eigener Temperaturschrank für den Kameraaufbau verwendet werden.

Alternativ können die drei Kameras durch drei Spiegel auf halber Distanz, die die drei Laserstrahlen auf nur eine Kamera abbilden, ersetzt werden. Dabei kann der Abstand der Spiegel zum Temperaturschrank noch weiter verkürzt werden je höher die Auflösung der Kamera ist. Der Vorteil des Verfahrens liegt hierbei nicht nur in der Platzersparnis, sondern auch in der erleichterten Auswertung, da man die relativen Ablenkungen direkt vergleichen kann. Einzelne Stahlen können nach Bedarf durch eine Blende abgedeckt werden. Der Nachteil bei diesem Ansatz ist die höhere Empfindlichkeit gegen Störungen, weil schon kleinste Verschiebungen der Spiegel das Messergebnis durch ungewollte Winkelabweichungen unbrauchbar machen.
Durch die geforderte hohe Präzision müssen die Spiegel ebenfalls von höchster Qualität sein, d.h. hohe Planität und hohe Oberflächengüte aufweisen. Besonders wichtig sind bei diesem Messverfahren die äußeren Einflüsse. Faktoren wie die Nähe einer Autobahn, Gebäudespannungen etc. müssen bei der Standortwahl einkalkuliert werden, weil Spiegel die Eigenschaft haben, Fehlwinkel zu verdoppeln. Da sich die gesamte Messung nur im µm-Bereich bewegt, haben bereits Fehlstellungen in der Größenordnung von wenigen Winkelsekunden merklichen Einfluss auf das Messergebnis.

Mit kleinen Änderungen, wie wasserdichten Wänden oder luftdichter Versiegelung kann man den Temperaturschrank mit einem zusätzlichen Luftbefeuchter und Kompressor zu einem Klimaschrank erweitern. Je nach zu vermessender Optik wäre auch ein Vakuumschrank denkbar.
An der Auswertung eines abgeänderten Verfahrens ändert sich allerdings nichts, da sie auf dem selben Messprinzip beruht.

Literaturverzeichnis

[1] pciGrabber-4plus Hardware-Manual, PHYTEC Technologie Holding AG, Mainz, 2001
[2] Temperaturmesssystem für DALLAS 1820, Hygrotec Messtechnik, Titisee-Neustadt, 2000
[3] Kurzanleitung für die grado-Regler 913, 923, Hengstler GmbH, Aldingen, 1998
[4] Installations- und Betriebsanleitung für Industrieregler grado913 und grado923, Hengstler GmbH, Aldingen, 1997
[5] SCIA 485 - Interface Installations- und Betriebsanleitung, Hengstler GmbH, Aldingen
[6] Gebrauchsanleitung zum HM-485 Schnittstellenwandler, Connectivity Expert, Texas
[7] Introduction to RS 422 & RS 485, Hardware Server http://www.hw.cz/english/docs/rs485/rs485.html, 1997/98