

Diplomarbeit

Ansteuerung eines Mikroskops zur Untersuchung von Mikrofossilien



Studenten: Andreas Herzig, Lukas Schmutz
Dozent: Prof. Dr. Jean Eisenecker
Expertin: Martine Lefebvre
Auftraggeber: Naturhistorisches Museum Basel, Dr. Michael Knappertsbusch

10. Januar 2007

Abstract

Diese Dokumentation entstand im Rahmen der Diplomarbeit an der Fachhochschule Nordwestschweiz. Auftraggeber ist das Naturhistorische Museum Basel und Projektbetreuer ist Prof. Dr. Jean Eisenecker.

Ziel dieser Diplomarbeit ist es, künftig die Ausrichtung von Mikrofossilien nicht mehr manuell, sondern automatisch, vorzunehmen. Dazu wird ein bestehender Aufbau verwendet. Er umfasst ein Mikroskop mit Beleuchtung und Kamera, ein Schwenktisch und einen Steuerrechner.

Das bisherige Vorgehen der Mitarbeiter und Mitarbeiterinnen des Naturhistorischen Museums bei der Vermessung von Mikrofossilien war, mit Hilfe von Videokameras und bildverarbeitender Software jedes Objekt unter dem Mikroskop von Hand zu justieren, abzubilden, digital aufzubereiten und abzuspeichern.

Die Automation dieser Abläufe wird erreicht, indem der PC an Hand des Kamerabildes automatisch fokussiert und dem Schwenktisch mitteilt, wie er das Objekt auszurichten hat. Die Software wird mit LabView 8.20 realisiert.

Inhaltsverzeichnis

1	PROJEKTÜBERSICHT	5
2	AUFGABENSTELLUNG	6
2.1	PROJEKT	6
2.2	AUFGABEN	6
2.3	ZUSATZBEDINGUNGEN	6
3	MIKROFOSSILIEN	7
4	AUFBAU	9
4.1	MIKROSKOP	10
4.2	SCHWENKTISCH	10
4.3	BELEUCHTUNG	11
5	VORGEHEN	12
5.1	ARBEITSAUFTEILUNG	12
5.2	VERWENDETE SOFTWARE	12
5.3	SCHWERPUNKTE DER ARBEIT	12
5.4	VERMUTETE PROBLEME	13
5.5	ZEITPLAN	14
5.6	EVALUATION BILDERFASSUNG	15
6	BILDERFASSUNG	16
6.1	HARDWARE	16
6.2	SOFTWARE	17
6.3	BILDVERARBEITUNG	17
6.4	BILD SPEICHERN	21
6.5	LEUCHTQUELLE	22
7	MOTORFOKUS	23
7.1	HARDWARE	23
7.2	PROTOKOLL	23
7.3	IMPLEMENTATION IN LABVIEW	24
8	AUTOFOKUS	25
8.1	EVALUIERUNG	25
8.2	STANDARDABWEICHUNG VOM HISTOGRAMM	26
8.3	GRAUSTUFENVERTEILUNG ENTLANG EINER LINIE	28
8.4	FLÄCHENMESSUNG	29
8.5	INTERPOLATION	29
8.6	ROI (REGION OF INTEREST)	30
8.7	VERZÖGERUNG	31
9	AUTOMATISCHES AUSRICHTEN	32
9.1	AUTOCENTER	32
9.2	AUTOTILT	33
9.3	AUTOROTATE	35
9.4	AUTOSHIFT	36
10	SOFTWARE	37
10.1	PROJEKTGLIEDERUNG	38
10.2	MAIN	44
10.3	AUTOMATIKMESSUNG (SPNLAUTO.VI)	46
10.4	EINZELMESSUNG (SPNLSINGLE.VI)	48
10.5	AUFBAU DER STATEMACHINE	50
10.6	FUNCTIONAL GLOBAL VARIABLE	51
10.7	TYPEDEFINITIONEN	55
10.8	EVENTSTRUKTUR	55

10.9	FEHLERCODES	57
11	ERFAHRUNGEN	58
11.1	UMWEGE UND PROBLEME	58
11.2	ERREICHTE ZIELE	58
11.3	FEEDBACK.....	59
11.4	AUSBLICK.....	60
12	DANKSAGUNG	61
13	ERKLÄRUNG	62
14	QUELLENVERZEICHNIS	63
14.1	LINKLISTE	63
14.2	SEMESTERARBEITEN	63
14.3	LITERATUR	63
15	ANHANG	64

1 Projektübersicht

Um die in diesem Dokument beschriebene Arbeit und ihre Entwicklung zu verstehen, ist ein gewisses technisches Grundwissen, sowie fundierte Kenntnisse in LabView, erforderlich. Sie ist für Personen gedacht, welche sich intensiver mit dem Thema beschäftigen oder diese Arbeit weiterentwickeln wollen.

Auftraggeber ist das Naturhistorische Museum Basel, welches Untersuchungen an Mikrofossilien vornimmt. Diese Fossilien stammen vom Meeresboden und werden durch das internationale Ocean Drilling Projekt zu Tage gebracht. In der Schweiz wird dieses Projekt vom Schweizerischen Nationalfond getragen. Es soll die klimatischen Verhältnisse und die Evolution anhand dieser Mikrofossilien zu jener Zeit aufzeigen.

Vor unserer Diplomarbeit wurde schon eine Semesterarbeit für dieses Projekt gemacht. Diese Arbeit umfasste einem Kipp-Schwenktisch, der auf einem X/Y-Tisch montiert ist. Durch diese Konstruktion muss man aber nach jedem Kippen das Bild wieder nachführen, was sehr mühsam und zeitaufwändig ist. Deshalb kam der Wunsch von Herrn Dr. Knappertsbusch einen Tisch zu konstruieren, bei welchem der X/Y-Tisch kipp- und schwenkbar ist. Zudem soll das Ganze automatisiert werden, so dass jedes einzelne Fossil ausgerichtet und fotografiert werden kann.

Der 4-Achsen-Tisch wurde von Herrn Binggeli in Brugg / Windisch gefertigt. Er schrieb auch das ganze Interface für LabView, so dass wir den komplett funktionierenden Tisch übernehmen konnten.

Mit der Übergabe des Schwenktisches begannen wir mit unserer Arbeit. Als erstes wurde der Treiber zum Schwenktisch an unsere Anforderungen angepasst. Auch wurde schon zu Beginn die Firma *Leica* wegen des Motorfokus angefragt, da der Autofokus der erste Schritt der Bildauswertung darstellt. *Leica* lieferte uns eine umfangreiche Dokumentation über ihren Motorfokus, so dass wir diesen ohne Probleme in LabView implementieren konnten. Die Bilderfassung unter LabView stellte uns anfangs vor gewisse Herausforderungen, welche wir jedoch ohne grösseren Zeitverlust meisterten. Nachdem die Bilderfassung funktionierte, konnten die Softwareteile für das automatische Ausrichten und Fokussieren geschrieben werden. Parallel dazu wurde das Hauptprogramm entwickelt, welches auf einer Statemachine basiert.

Während dieser Zeit kamen wir sehr schnell vorwärts, da wir einerseits bessere Dokumentationen von *Leica* bekommen hatten als erwartet, und andererseits wir fast keine Probleme mit der Implementation in LabView hatten. Es kamen weitere Ergänzungen zur ursprünglichen Aufgabenstellung dazu. Diese konnten wir weitgehend berücksichtigen, da wir sehr gut im Zeitplan lagen.

Gegen Ende nahm die Optimierung der kompletten Software mit dem ganzen Error-Handling noch recht viel Zeit in Anspruch. Auch die Inbetriebnahme der Software und Hardware auf dem Zielrechner ohne LabView-Entwicklungsumgebung verlief nicht ganz ohne Probleme.

Unsere Diplomarbeit beinhaltet Aufgabenstellungen, die uns auf verschiedenen Ebenen herausforderten: entwickeln einer adäquaten Software, konstruieren eines funktionsfähigen Objektes, kommunizieren und zusammenarbeiten im Partnerteam und mit dem Auftraggeber. Es ist uns gelungen, dem Naturhistorischen Museum einen ausgereiften Aufbau zum Untersuchen von Mikrofossilien zu übergeben. Mit dieser Anlage wird Herr Dr. Michael Knapperstbusch in Zukunft viel Zeit sparen und seine Forschungen so vorantreiben können.

2 Aufgabenstellung

Klasse: Diplomklasse
Kandidaten: Andreas Herzig, Lukas Schmutz
Titel: Ansteuerung eines Mikroskops zur Untersuchung von Mikrofossilien

2.1 Projekt

Am Naturhistorischen Museum in Basel werden Mikrofossilien mit Mikroskop und Kamera untersucht. Mit der Bestimmung des Formfaktors können Aussagen über die Entwicklung der Mikrofossilien und den zurzeit herrschenden klimatischen Verhältnissen gemacht werden. Die Mikrofossilien befinden sich in einer Halterung, die auf einem Schwenktisch befestigt ist. Der heutige manuelle Messvorgang besteht aus Fokussierung des Mikroskops, Orientierung des Mikrofossils mit dem Schwenktisch, so dass es senkrecht steht, ev. neue Fokussierung, Bildaufnahme, Archivierung und Bildauswertung. Künftig soll die Mikrofossilienausrichtung automatisch erfolgen. In dieser Arbeit soll die Fokussierung des Mikroskops und die Ausrichtung des Schwenktisches automatisiert werden.

2.2 Aufgaben

- Untersuchung der Schnittstelle des *Leica*-Mikroskops
- Realisierung der Mikroskop-Ansteuerung mit Labview und entsprechender Schnittstelle
- Evaluation verschiedener Verfahren zur automatischen Fokussierung mit und ev. ohne Bildverarbeitung.
- Realisierung der automatischen Fokussierung des *Leica*-Mikroskops mit Labview mit Einbindung der Bilderfassung mit der Mikroskopkamera.
- Positionierung des Schwenktisches, um das Mikrofossil in vertikale Lage zu bringen.
- Realisierung der ganzen Einheit, um automatisch jedes Mikrofossil der Halterung unter das Mikroskop zu schieben, es zu fokussieren und vertikal auszurichten und ein Foto aufzunehmen
- Erstellung einer Bedieneroberfläche

2.3 Zusatzbedingungen

Während der ganzen Arbeit ist ein Laborjournal zu führen. Zu Beginn der Arbeit ist ein Terminplan aufzustellen.

Arbeitsbeginn: Dienstag, 24. Oktober 2006, 8.30 Uhr
Abgabetermin: Mittwoch, 10. Januar 2007, 17.00 Uhr
Präsentation: Freitag, 12. Januar 2007, 16.15 Uhr
Ausstellung: Dienstag, 16. Januar 2007, 15.00- 21.00Uhr

Examinator: Prof. Dr. Jean Eisenecker
Experte: Martine Lefebvre

3 Mikrofossilien

Die Erdneuzeit (von vor 65 Millionen Jahren bis heute) ist ein Zeitbereich, in welchem sich die Ozeane, ihre Strömungsmuster, und ihre vertikale Struktur stark verändert haben. Plattentektonische Veränderungen, abwechselnde Warm-, Kalt- und Eiszeiten, sowie die Entstehung der Eiskappen in der Antarktis (vor etwa 33-34 Millionen Jahren) und in der Arktis (Beginn hauptsächlich vor etwa 3,5-3 Millionen Jahren) führten zu zeitlich abgestuften, weltweiten Umschichtungen und Umverteilung von ozeanischen Wassermassen. Die dadurch entstandenen neuen marinen Ökosysteme und Nischen beschleunigten nicht nur die Anpassung und Evolution zahlreicher planktonischer Lebewesen, sondern führten auch zum Aussterben vieler Arten. Insbesondere die kalk- und kieselschalenbildenden, einzelligen Lebewesen, welche die oberen Wasserschichten der Meere bevölkern, sind ideale Untersuchungsobjekte dieser Vorgänge. Ihre Schalen finden sich in ungeheuerlich grosser Anzahl in den Ablagerungen der Meere wieder. Die Veränderlichkeit dieser Schalen über die geographischen Breiten und über Jahrtausende der Entwicklungsgeschichte hinweg bieten einerseits ein fantastisches Archiv weltweiter Klimaveränderungen und andererseits ein natürliches Laboratorium, um die Ursachen und Pfade der Evolution besser zu verstehen. Ein solches Verständnis ist von grosser Wichtigkeit, denn Evolution ist der zentrale Prozess der Artenbildung und somit die Wiege für Artenvielfalt.

In diesem Zusammenhang wurden die planktonische *Foraminiferenart Globorotalia menardii* (= planktonisch lebendes, einzelliges Tierchen mit einer Kalkschale) ausgewählt, um an ihrem Beispiel die Vorgänge der Artenbildung zu untersuchen. Aus Bohrkernen gewonnene Proben werden die Schalen extrahiert und diese mit Hilfe spezieller Computerprogramme vermessen. Bisher wurden die Schalen aus zwei Bohrlöchern im Detail untersucht: Eines befindet sich im *Columbia Becken* in der Karibik, das andere ist im äquatorialen Ostpazifik. Diese Orte erlauben uns die Entwicklungsgeschichte im *Globorotalia menardii* Artenkomplex aufzuschlüsseln und den besonderen Einfluss der Schliessung der ehemaligen Meeresverbindung zwischen Karibik und Pazifik vor etwa 2-3 Millionen Jahre zu untersuchen.

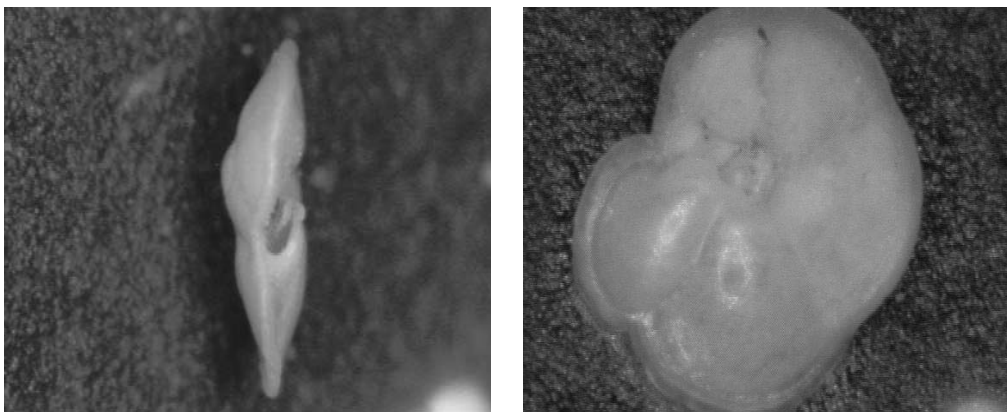


Abb. 1 Mikrofossil in Kiel- und Spiralansicht

Zur detaillierten Vermessung von Mikrofossilschalen müssen diese einzeln auf einem Objektträger in Reih und Glied von Hand fixiert und vororientiert werden.

Eigens vom Museum entwickelte Computerprogramme ermöglichen es, die Umriss der Mikrofossilien aus dem Gesamtbild als kartesische Koordinaten zu extrahieren. Die gewonnenen Daten werden mit einer weiteren Reihe von Spezialprogrammen analysiert und davon morphologische Kenngrößen der Schalen abgeleitet. Erst dann wird es möglich, mit Hilfe graphischer und statistischer Analyse die Muster der Evolution der Schalen zu erkennen.

4 Aufbau



Abb. 2 Aufbau

Der ganze Aufbau besteht aus folgenden Teilen:

- Mikroskop
- 4-Achs-Tisch
- Kamera
- Beleuchtung
- PC mit Interface-Karten

Der Schwenktisch wird mit zwei Passstiften auf dem Fuss des Mikroskops fixiert. Beleuchtet wird entweder über die Ringbeleuchtung, die unter dem Objektiv fixiert ist, oder über Schwannenhäule, welche man mit Polarisatoren ergänzen kann.

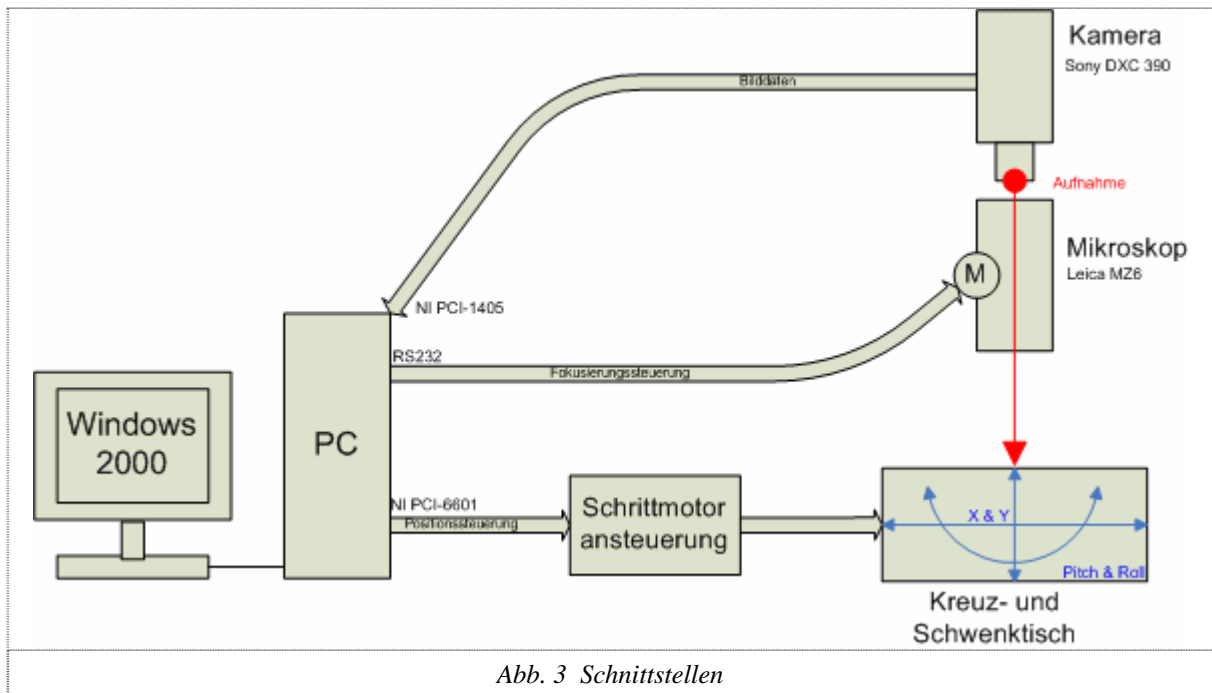


Abb. 3 Schnittstellen

4.1 Mikroskop

Das Mikroskop *Leica MZ6* besitzt neun verschiedene Vergrößerungsstufen (0.63 – 4.0) und eine C-Mount-Schnittstelle für eine Kamera. Die Blende kann auch verstellt werden, was sich dann stark auf den Treshold der Bildauswertung auswirkt. Das Mikroskop ist zusätzlich noch mit einem Motorfokus ausgestattet. Dieser lässt sich einerseits mit einem Handrad, welches über CAN verbunden ist, oder andererseits über die RS232 mit einem PC steuern. In diesem Projekt wird der Motorfokus über die serielle Schnittstelle angesprochen.

4.2 Schwenktisch

Der komplette Schwenktisch und seine Ansteuerung wurden von Herrn Daniel Binggeli der FHNW in Brugg realisiert.

Der 4-Achsen-Schwenktisch besitzt folgende Achsen; X, Y, Roll und Pitch. Der X/Y-Tisch ist kardanisch aufgehängt, so dass beim Schwenken keine Verschiebungen entstehen. Die einzelnen Achsen werden mit Schrittmotoren angesteuert, welche mit einem Untersetzungsgetriebe ausgestattet sind. Der Tisch kann sich um 30° neigen und jeweils 50mm in X- und Y-Richtung fahren. Die vier Schrittmotorenverstärker und das Netzteil sind in einer externen Steuerbox untergebracht.

4.3 Beleuchtung

Die Beleuchtung besteht aus einer Halogenlichtquelle mit 120 Watt, welche in einem externen Gehäuse untergebracht ist. Daran lässt sich ein Glasfaserkabel anschliessen, welches das Licht zum Mikroskop leitet. Hier kann man entweder die Ringbeleuchtung, welche unter dem Objektiv befestigt wird und das Licht gleichmässig verteilt, verwenden, oder die Schwanenhäse, welche man zusätzlich mit Polarisatoren versehen kann. Durch das polarisierte Licht lässt sich der Hintergrund besser ausblenden. Zusätzlich lässt sich die Leuchtstärke einstellen.

5 Vorgehen

5.1 *Arbeitsaufteilung*

Die komplette Bildverarbeitung inklusiv Fokusansteuerung wurde von Andreas Herzig realisiert. Lukas Schmutz war für das ganze Hauptprogramm verantwortlich. Die detaillierte Arbeitsaufteilung ist im Zeitplan ersichtlich

5.2 *Verwendete Software*

Entwicklungs-PC

Pentium 4, SXGA (1280x1024), Windows XP SP2

Ziel-PC

Pentium 4, SXGA (1280x1024), Windows 2000

Entwicklungsumgebung

National Instruments LabView 8.20

Dokumentation

Die komplette Dokumentation wurde mit Microsoft Word 2003 erstellt und ins pdf-Format konvertiert.

Treiber

Bezeichnung	Beschreibung	Version
NI-IMAQ	Bilderfassung	3.7
NI-DAQmx	Counterkarte	8.3
NI-Serial	COM-Port	3.1
NI-RunTime	LabView 8.2 RunTime-Engine	8.2
NI-Vision-RunTime	RunTime für Vision Komponenten	8.3

5.3 *Schwerpunkte der Arbeit*

- anpassen des Schwenktischtreibers
- erfassen und verarbeiten des Videobildes unter LabView
- ansteuern des Motorfokus über RS232
- automatisches Fokussieren
- automatisches Ausrichten der Fossilien
- abspeichern der Bilder
- komplette Software, welche die Steuerung der einzelnen SubVI's übernimmt

5.4 Vermutete Probleme

Die Ansteuerung des Motorfokus über die serielle Schnittstelle war für uns eine grosse Unbekannte. Wir konnten am Anfang den Aufwand für diesen Arbeitsteil nicht abschätzen, da wir nicht wussten, wie kooperativ *Leica* sein würde. Daher rechneten wir mit zwei Wochen im Zeitplan. Wir bekamen jedoch von *Leica* den kompletten Befehlssatz, was uns sehr schnell voranbrachte. Durch diesen Vorsprung von zwei Wochen waren wir nun flexibel, um zusätzliche Funktionen in die Software einzubauen.

Die Thematik *Autofokus* war zusätzlich ein zentraler Punkt unseres Projekts, da alle weiteren Schritte der Bildverarbeitung ein scharfes Bild benötigten. So musste eine vernünftig schnelle und vor allem realisierbare Methode gefunden und umgesetzt werden.

5.5 Zeitplan

24.10.2006	30.10.2006	06.11.2006	13.11.2006	20.11.2006	27.11.2006	04.12.2006	11.12.2006	18.12.2006	03.01.2007	10.01.2007
------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------

Kalenderwoche

43	44	45	46	47	48	49	50	51	1	2
----	----	----	----	----	----	----	----	----	---	---

Leica-Schnittstelle	Einarbeiten			X									
	Kommunikation mit LabView			X									
LabView	Senden über RS232	X											
	Ansteuern des Mikroskops			X									
	Automatisches Fokussieren			X	X	X							
	Bedienoberfläche			X	X	X	X						
	Automatisches Anfahren der Felder		X	X				X					
	Automatisches Zentrieren				X								
	Automatisches Ausrichten				X	X							
	Hauptprogramm			X	X	X	X						
	Statemachine			X	X								
	Inbetriebnahme Hauptprogramm						X	X					
	Fehler Ausbessern						X	X	X				
	Inbetriebnahme auf Fremdrechner						X		X				
	Fokussierung	Einarbeiten	X										
Evaluation verschiedener Verfahren			X										
Ausarbeiten eines Verfahrens			X										
Schwenktisch	Einarbeiten	X											
	VI anpassen		X										
Dokumentation	Schreiben	X					X	X	X	X	X	X	X

Meilensteine

			M1	M2	M3		M4	M5		M6
--	--	--	----	----	----	--	----	----	--	----

Legende:

soll	ist	
	X	Schmutz
	X	Herzig
	X	Beide

- M1 automatisches Fokussieren
- M2 automatisches Anfahren der Felder
- M3 automatisches Ausrichten
- M4 Hauptprogramm läuft
- M5 Programm läuft auf Fremdrechner
- M6 Abgabe

5.6 Evaluation Bilderfassung

Anfangs war geplant, dass man die digitale Kamera *JVC KY-F75U* mit Firewire-Schnittstelle verwendete, da man diese an einem IEEE 1394-Port am PC anschliessen konnte. Nach dem Ausprobieren und Nachschlagen auf der Homepage von *National Instruments*, mussten wir feststellen, dass LabView eine eigene 1394-Karte benötigte, die vom IMAQ-Treiber unterstützt wurde. Diese Karte kostete jedoch 700 CHF, was das Budget überschreiten würde. Da noch eine analoge Framegrabber-Karte von der letzten Semesterarbeit vorhanden war, beschlossen wir, diese Karte in Verbindung mit der *Sony DXC 390P* einzusetzen.

6 Bilderfassung

6.1 Hardware

Kamera

Für die Bilderfassung wird die *Sony DXC 390P* mit 3CCD Chip verwendet, die direkt mit der optischen C-Mount Schnittstelle mit dem Mikroskop verbunden ist. Dies ist eine Kamera mit digitaler Signalverarbeitung, aber mit analogem BNC- und RGB-Ausgang. Sie liefert ein Bild mit einer Auflösung von 640 x 480 Pixel und hat einen externen Triggeranschluss.



Abb. 4 Sony DXC 390 3CCD Kamera

Framegrabber-Karte

Die Digitalisierung der analogen Bilder übernimmt die PC-1405 Framegrabber-Karte von *National Instruments*. Diese Karte unterstützt Monochrom- und Farbbilder und besitzt einen externen Triggerausgang. Es kann sogar ein ROI (Region of Interest) programmiert werden, der dann direkt von dieser Karte umgesetzt wird.

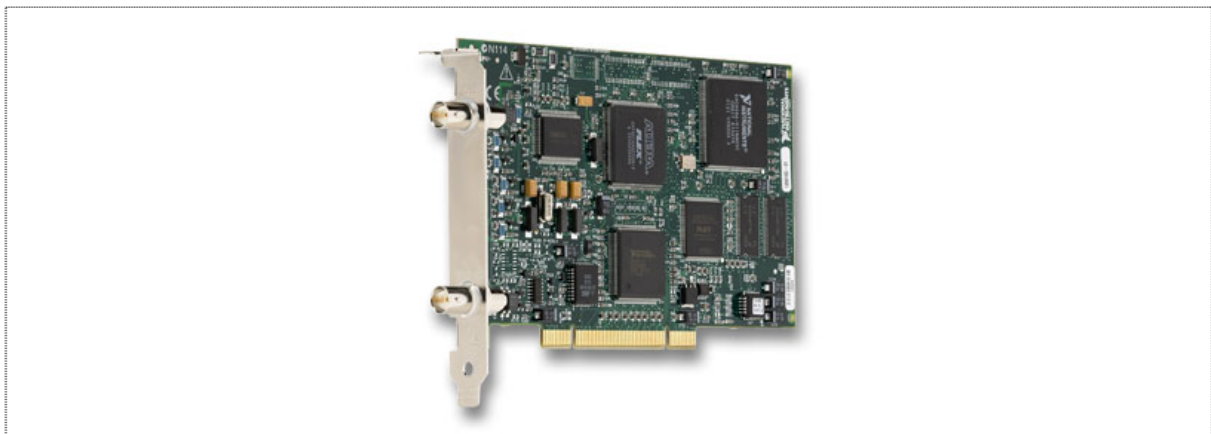


Abb. 5 PCI-1405 von National Instruments

6.2 Software

Da *National Instruments* die firmeneigenen Framegrabber-Karten sehr gut unterstützt, ist es ohne grossen Aufwand möglich, ein Bild über den IMAQ-Treiber in LabView darzustellen. Die Bildauswertung gestaltet sich aber um einiges schwieriger. Mit dem *Vision Assistant* ist zwar die Verarbeitung problemlos möglich, dieser verarbeitet aber keine Kamerabilder, sondern nur abgespeicherte Bilddateien.

Auch der *Vision Builder*, der laut *National Instruments* für solche Anwendungen ausgelegt ist, hat seine Schwachstelle. Er erlaubt zwar den Zugriff auf Kamerabilder, kann aber nicht alle Bildbearbeitungsfunktionen bewerkstelligen. So werden die Codeanteile der Verarbeitungskette vom *Vision Assistant* erstellt und über den *Vision Builder* wird der Kamerazugriff geregelt.

Diverse Bausteine müssen nun mit Parametern versehen werden, z.B. die Kanalnummer der Kamera oder der Bildtyp für die Framegrabber-Karte.

Leider liefert die Kamera ein 640 x 480 Pixel-Bild, das einen 10 Pixel breiten, schwarzen Rand hat. Da für die Bildverarbeitung dieser Rand sehr störend ist, wird das Bild nur auf 630 Pixel abgetastet. Jetzt wird das Bild in ein Graustufenbild umgewandelt, da es später nur so verwendet werden kann.

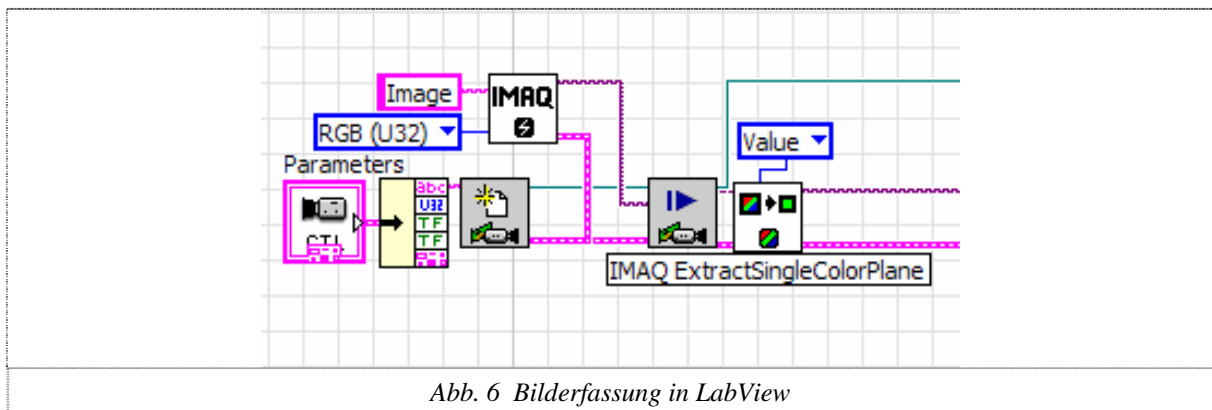


Abb. 6 Bilderfassung in LabView

6.3 Bildverarbeitung

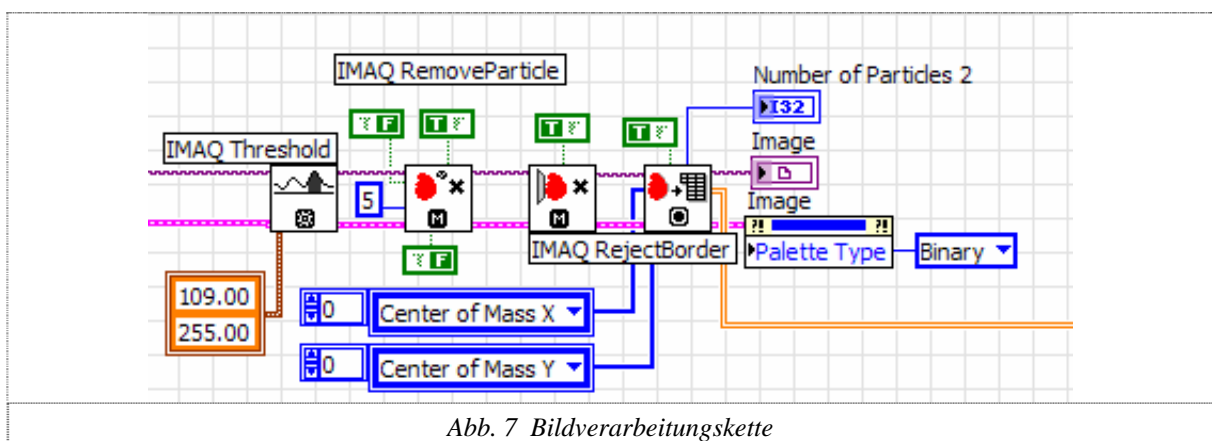
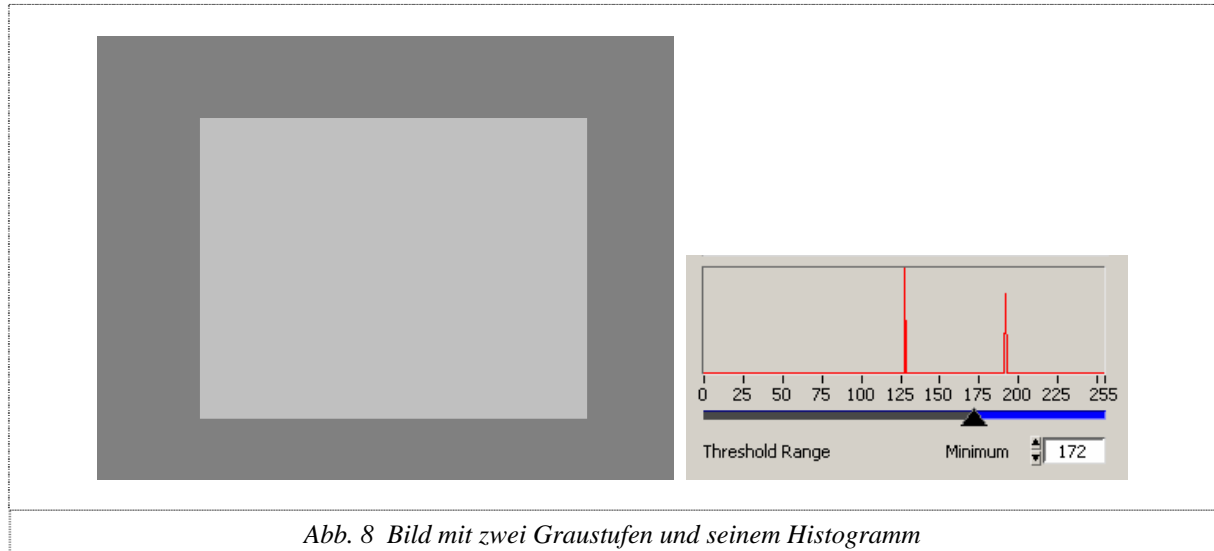


Abb. 7 Bildverarbeitungskette

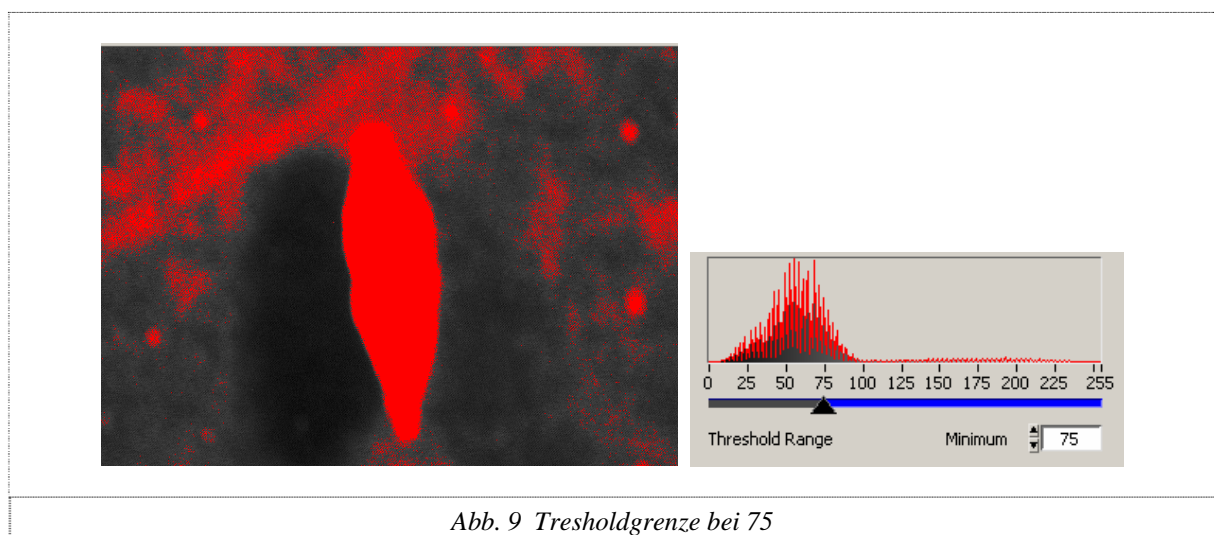
Threshold

Für die weitere Bildverarbeitung wird in diesem Projekt meistens ein Binärbild benötigt, welches mit einem Threshold erzeugt wird. Hat man klare Helligkeitsunterschiede zwischen dem Hintergrund und dem Objekt, kann man den Threshold gut automatisch erzeugen. In der Abbildung unten kann der Thresholdbereich zwischen 130 und 185 variiert werden, ohne dass sich das Binärbild verändert.



Besitzt jedoch das Bild mehr als zwei Graustufen, was in der Realität meist der Fall ist, ist es schon schwieriger die optimale Grenze für den Threshold zu finden. Bei unseren Bildern besitzt der Hintergrund viele Graustufen und ist auch leicht verschwommen. Daher ist es sehr schwierig, die optimale Thresholdgrenze automatisch zu finden. Aus dem Histogramm kann man erkennen, dass sich das Objekt fast nicht vom verschwommenen Hintergrund abhebt. Fast alle Graustufen befinden sich im Bereich von 15-100, was keine klare Grenze ergibt. Aus diesem Grund wird der Grenzwert anhand von vielen Versuchen mit der Ringbeleuchtung und dem polarisiertem Licht bestimmt. So erhalten wir die besseren Resultate als mit einem automatischen Thresholding.

Zudem hat die Blendeneinstellung einen grossen Einfluss auf die optimale Thresholdgrenze. Deshalb wird der Benutzer beim Aufstarten des Programms zum Einstellen der Blende aufgefordert.



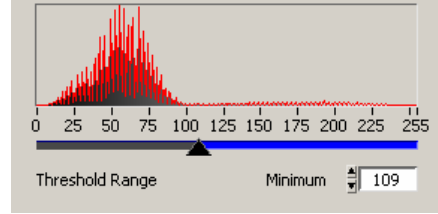
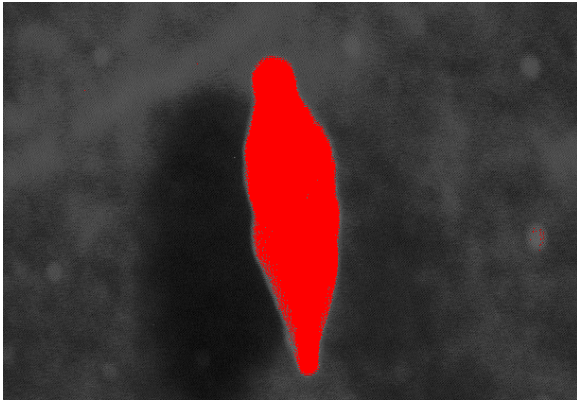


Abb. 10 Tresholdgrenze bei 109

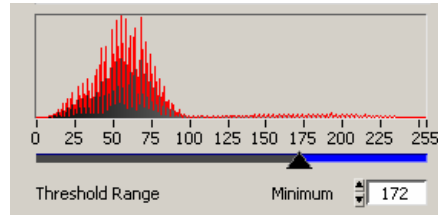
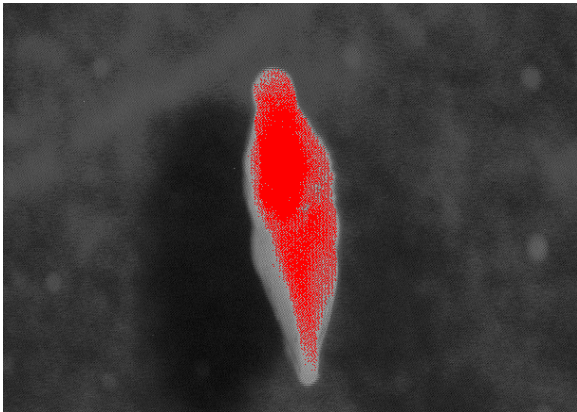


Abb. 11 Tresholdgrenze bei 172



Abb. 12 Binärbild

Reject Border

Um die Linien und Zahlen des Messprobenhalters wegzubekommen, wird *RejectBorder* verwendet, der alle Partikel entfernt, die den Rand berühren. Darum ist es auch wichtig, dass man keinen schwarzen Randstreifen der Kamera hat, sonst würden diese Objekte den Rand nicht berühren und somit nicht entfernt werden.

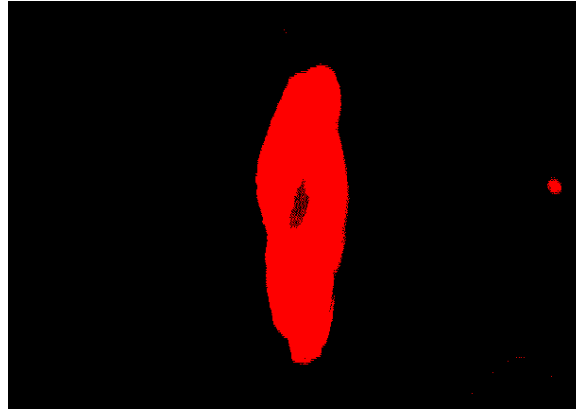


Abb. 13 *Reject Border*

Remove Particle

Als nächstes werden mit der Funktion *RemoveParticle* unerwünschte Staubkörner entfernt. Dieser Vorgang wird mehrmals wiederholt, normalerweise fünfmal. Wird immer noch mehr als ein Partikel erkannt, wird die Wiederholung so lange erhöht bis nur noch ein Partikel sichtbar ist.

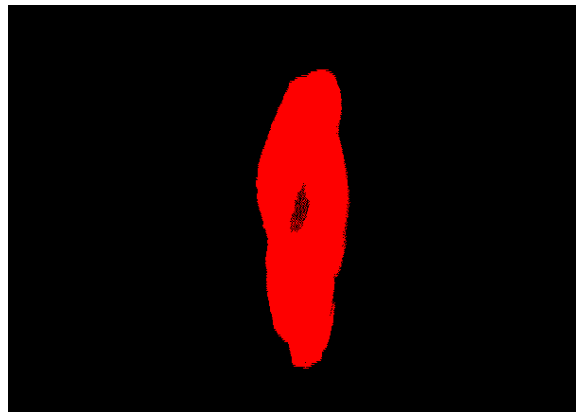


Abb. 14 *Remove Particle*

Fill Holes

Um eine geschlossene Fläche des Fossils zu erhalten, wird *Fill Holes* angewendet. Diese Funktion füllt die Löcher in den Objekten. Dies ist nötig, da die Ausrichtung zum Teil auf der Fossilfläche beruht.

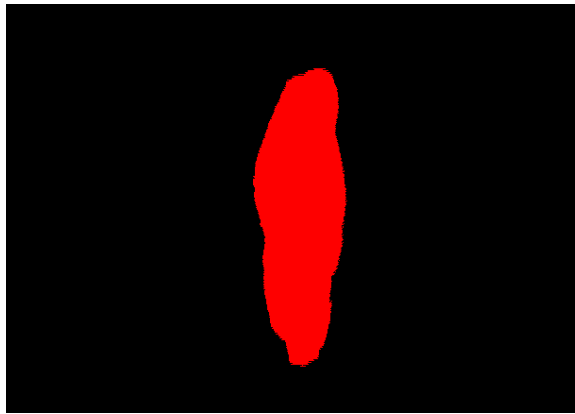


Abb. 15 Fill Holes

Partikel Analyse

Am Ende wird mit der Partikelanalyse kontrolliert, ob wirklich nur ein Partikel erkannt worden ist. Die Partikelanalyse liefert zudem die entsprechenden Messresultate, wie z.B. Zentrumskoordinaten, Fläche für die Weiterverarbeitung, Koordinaten des umschliessenden Rechtecks.

6.4 Bild speichern

Nach der ganzen Verarbeitung muss das Bild in einem geeigneten Format gespeichert werden. Da unser Bild nur eine Breite von 630 Pixel hat, muss es auf 640 Pixel vergrößert werden, damit es für die Weiterverarbeitung verwendet werden kann. Damit das Bild wieder die ursprüngliche Auflösung bekommt, wird am Schluss mit dem SubVI *expandImage.vi* ein schwarzer Streifen von 10 Pixel dazugefügt. Dem *pictureSaver* wird der gewünschte Speicherort per Dateipfad übergeben und dieser speichert das aktuelle Bild in folgenden Formaten; JPEG, TIFF, PNG, BMP.

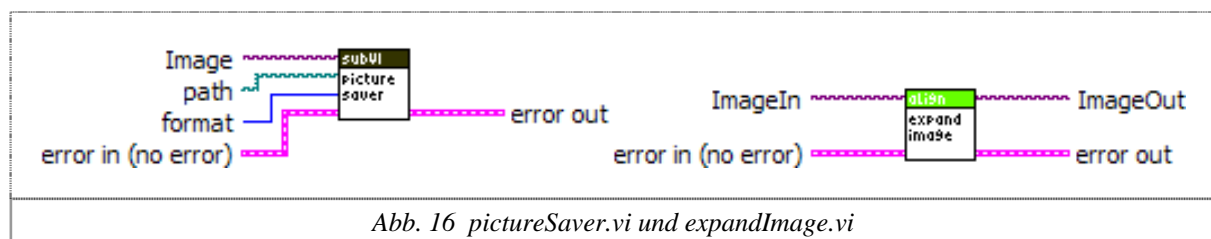


Abb. 16 pictureSaver.vi und expandImage.vi

6.5 Leuchtquelle

Die Ringbeleuchtung unterscheidet sich gross von der Schwanenhalsbeleuchtung mit Polarisatoren. Die Ringbeleuchtung leuchtet das gesamte Bild von oben aus, so dass keine Schatten entstehen können, aber der Hintergrund sehr dominant ist. Dies macht natürlich die Weiterverarbeitung schwieriger.

Die Schwanenhalsbeleuchtung muss man hingegen sehr gut ausrichten. Dafür erhält man durch das polarisierte Licht einen viel weniger dominanten Hintergrund. Problematisch sind bei dieser Beleuchtung aber stark nach aussen gekippte Fossilien am Sliderand. Hier können Schattenwürfe entstehen, die eine Weiterverarbeitung nahezu unmöglich machen.

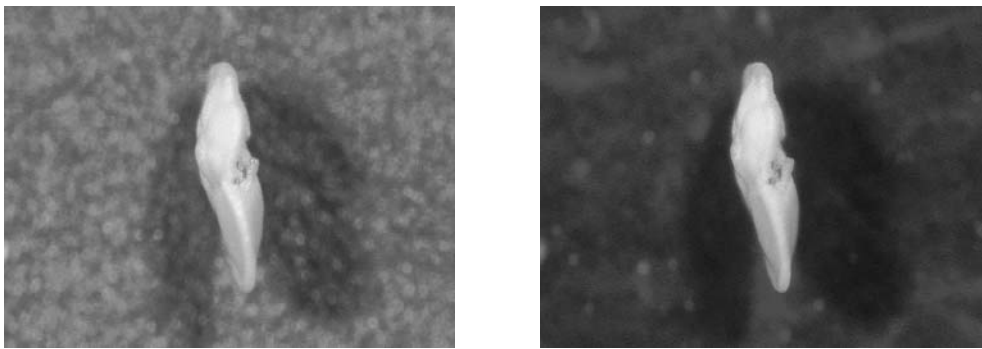


Abb. 17 ohne und mit polarisiertem Licht

7 Motorfokus

7.1 Hardware

Das *Leica* Mikroskop verfügt über einen motorisierten Fokus, der über ein Handrad, einen Fusstaster, oder über eine RS232-Schnittstelle angesteuert werden kann. In unserem Projekt muss die ganze Ansteuerung automatisiert werden. Somit ist die Anbindung an die serielle Schnittstelle unumgänglich. *Leica* lieferte uns die nötigen Unterlagen über den Befehlssatz. Der Motorfokus hat eine Auflösung von 1000 Schritten pro mm und kann absolut oder relativ fahren. Der Nullpunkt muss bei jedem Aufstarten wieder neu gesetzt werden. Der Fokus verfügt oben und unten über Endschalter, die auch per RS232 ausgelesen werden können.



Abb. 18 Motorfokus am Leica Mikroskop

7.2 Protokoll

Der Sendebefehl wird als normale ASCII-Zeichen gesendet und besteht aus folgenden Parametern:

Unit Identifier + Command Identifier + Parameters

Unit Identifier: 70 für Motorfokus-Einheit
 71 für Handrad

Command Identifier: ist der eigentliche Befehl, z.B. 031 bedeutet relatives Fahren

Parameters: gibt die Anzahl zu fahrende Schritte an

Der ganze Befehl wird mit *CarriageReturn* und *LineFeed* abgeschlossen.

Beispiel für das relative Abwärtsfahren von einem Millimeter:

```
70 031 -1000 <CR> <LF>
```

7.3 Implementation in LabView

Für das Senden über die serielle Schnittstelle wird das *serial.vi* verwendet. Dieses VI ermöglicht die Initialisierung und das Senden von Strings über die RS232 Schnittstelle. Das ganze VI beruht auf dem *SerialReadandWrite.vi*, welches von LabView zur Verfügung gestellt wird. Für unsere Anwendungen ist es bestens geeignet, da wir zusammengesetzte Strings an den Motorfokus verschicken.

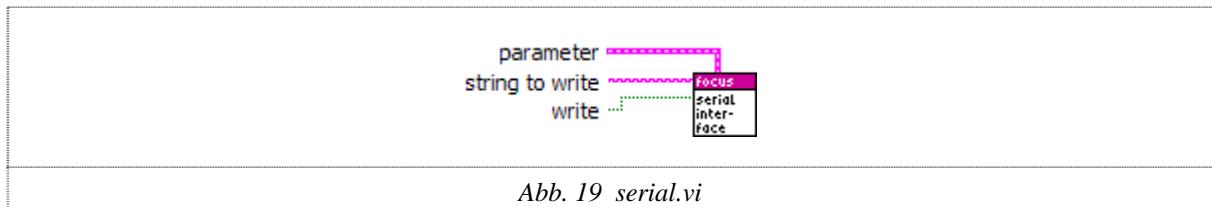


Abb. 19 *serial.vi*

Das *FocusControl.vi* wird hauptsächlich vom Autofokus als SubVI verwendet. Dieses VI ist nur für die Initialisierung und Ansteuerung des Fokus zuständig und kann nicht automatisch fokussieren. Es unterstützt das absolute und relative Bewegen des Fokus und überwacht auch die untere und obere Grenze um Kollisionen zu verhindern.

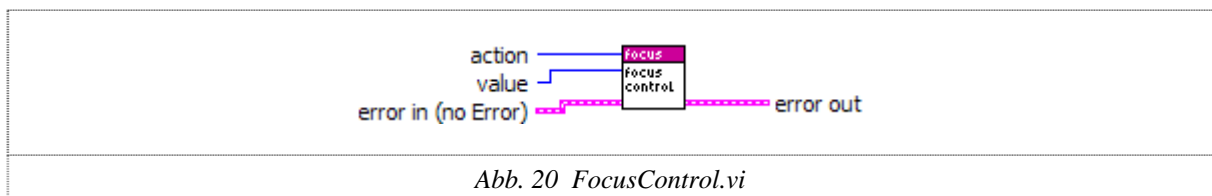


Abb. 20 *FocusControl.vi*

8 Autofokus

8.1 Evaluierung

In dieser Phase werden die verschiedenen Methoden zur automatischen Fokussierung genau auf die Realisierbarkeit und Zuverlässigkeit untersucht.

Kontrastmessung anhand des Histogramms

Das Bild ist am schärfsten, wenn der Kontrast am grössten ist. Bei dieser Methode wird die Bildweite des Objektivs solange variiert, bis der Kontrast maximal ist. Der Kontrast wird mittels der Standardabweichung vom Histogramm bestimmt.

Objekt mit kleinster Fläche

Wenn das Objekt die kleinste Fläche hat, ist seine Kontur auch am schärfsten. Sobald es unscharf wird, nimmt auch der Umfang und die Fläche zu. Diese Methode funktioniert nur, wenn das Bild schon grob fokussiert ist. Zusätzlich muss für diese Variante ein Binärbild erzeugt werden.

Frequenzmessung

Das Bild ist am schärfsten, wenn seine Frequenz am grössten ist. Durch Fouriertransformation lässt sich das Frequenzspektrum berechnen, dies erfordert aber hohe Rechenleistung. Da *Vision* keine Frequenzmessung unterstützt, wird diese Variante nicht weiter verfolgt.

Gradientenprofil

Es wird eine Gerade über das Objekt gelegt und entlang dieser die Steilheit der Schwarz-Weiss-Übergänge gemessen. Je steiler die Übergänge, umso schärfer das Bild. Diese Fokussiermethode liefert etwa die gleichen Ergebnisse wie die Standardabweichung vom Histogramm.

Methoden ohne Bildauswertung werden gar nicht in Betracht gezogen, da diese Varianten zusätzliche Hardware benötigen würden. Diese Mehrkosten sind vom Auftraggeber nicht erwünscht.

8.2 Standardabweichung vom Histogramm

Als erste Variante realisieren wir die Kontrastmessung anhand des Histogramms. Dazu wird von der Kamera ein Bild erfasst, das Histogramm gebildet und daraus die Standardabweichung berechnet.

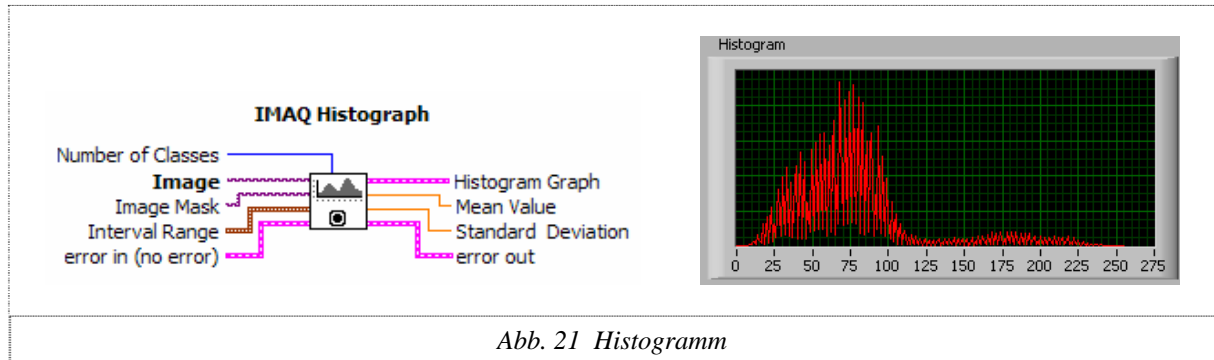


Abb. 21 Histogramm

Dort, wo die Standardabweichung maximal ist, hat das Bild den stärksten Kontrast. Bei polarisiertem Licht, welches den Hintergrund ausblendet, funktioniert diese Variante sehr gut. Wird jedoch die Ringleuchte mit unpolarisiertem Licht verwendet, kann der Hintergrund nicht ausgeblendet werden und es wird auf den Hintergrund scharf gestellt, da dieser dominanter ist.

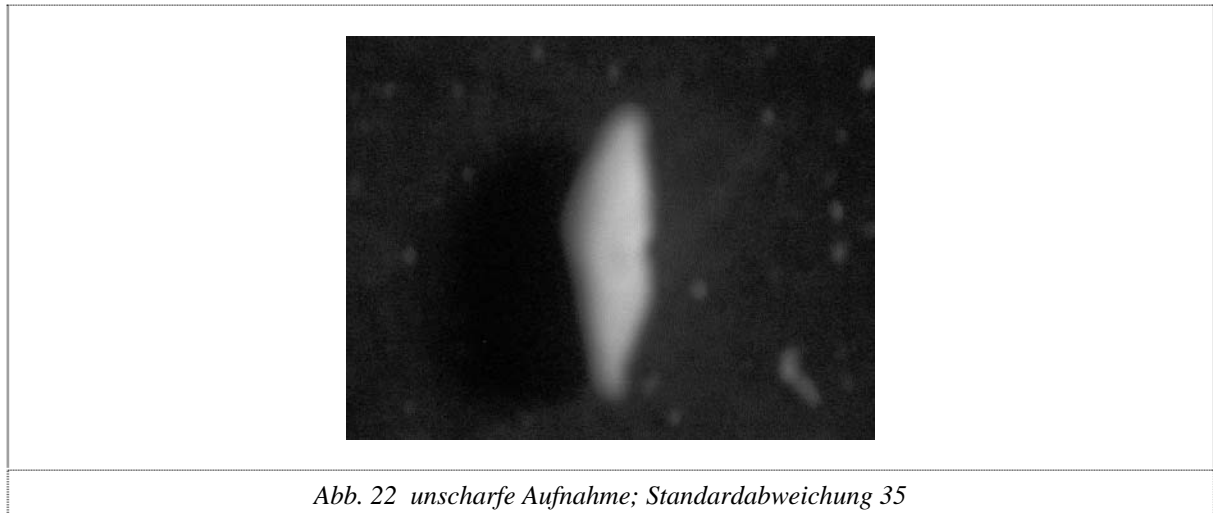


Abb. 22 unscharfe Aufnahme; Standardabweichung 35

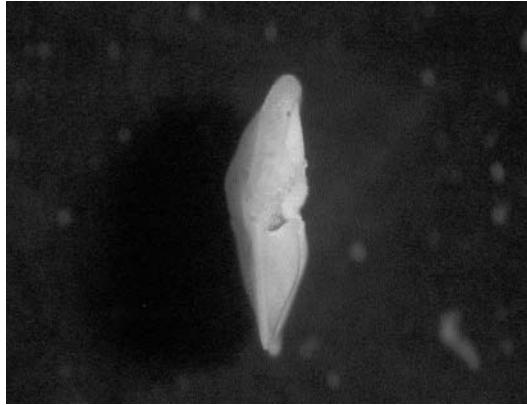


Abb. 23 scharfe Aufnahme; Standardabweichung 37

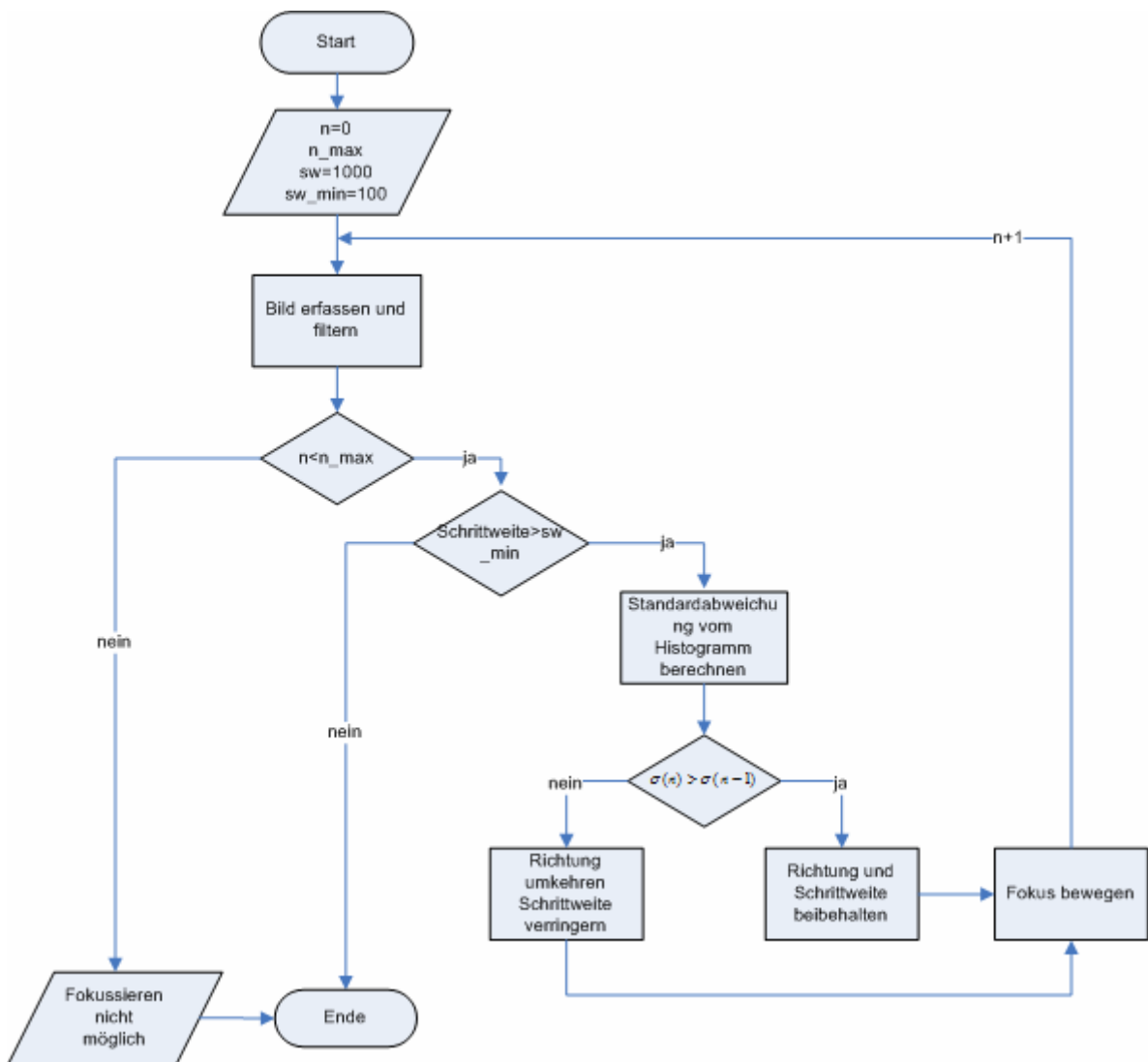
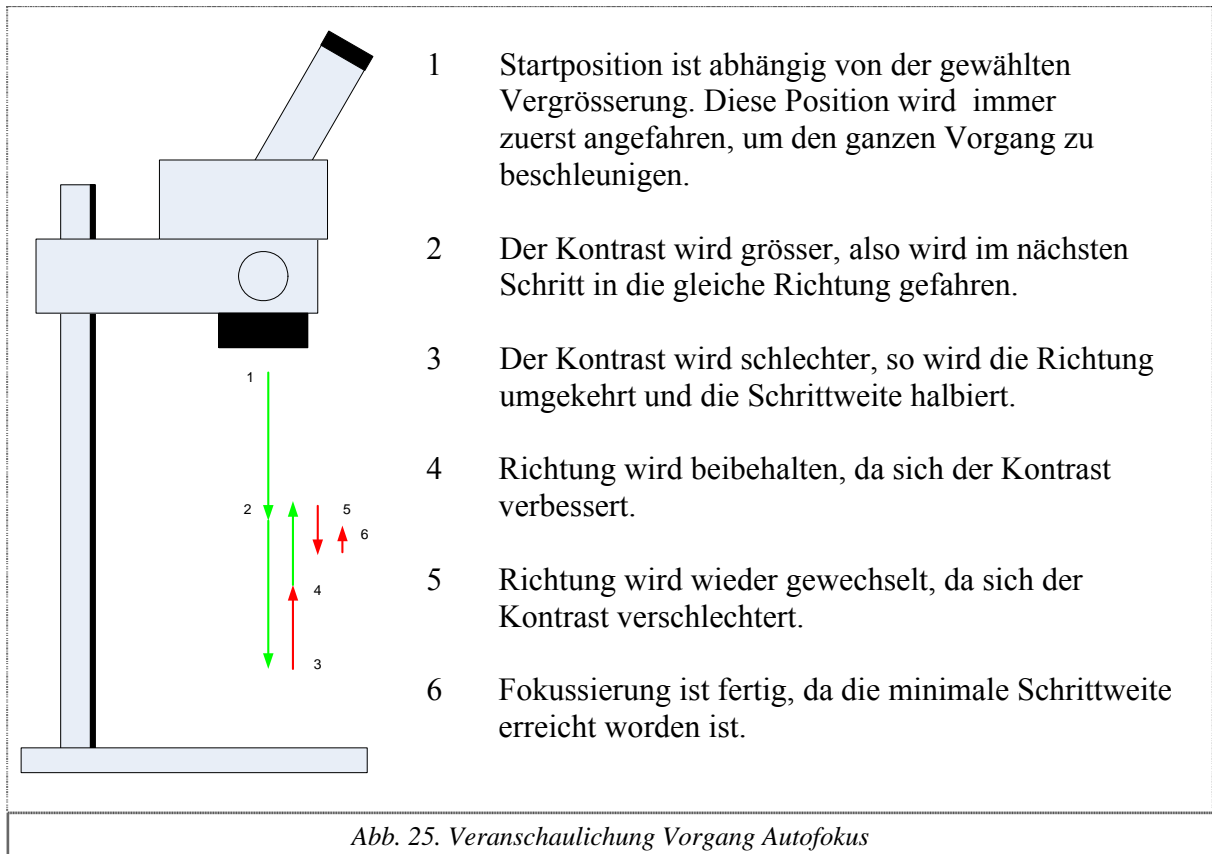
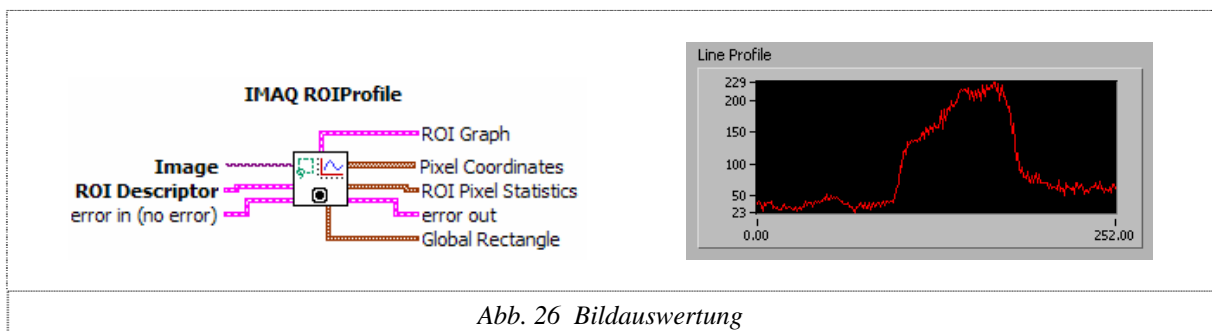


Abb. 24 Ablaufdiagramm Autofokus



8.3 Graustufenverteilung entlang einer Linie

Jetzt wird die Fokussierung anhand der Graustufenverteilung entlang einer Linie implementiert. Auch hier ergibt sich der schärfste Punkt bei der maximalen Standardabweichung. Da auch diese Variante indirekt auf der Kontrastmessung basiert, werden ähnliche Resultate wie mit dem Histogramm erzielt. Es wird aber immer noch auf den Hintergrund scharf gestellt, und somit zeigt sich keine Verbesserung.



8.4 Flächenmessung

Um auch scharfe Bilder mit unpolarisiertem Licht zu bekommen, wird die erste Variante mit der Flächenmessung kombiniert. So wird zuerst mit dem Kontrast grob fokussiert, um dann mit der Flächenmessung das Bild richtig scharf zu bekommen. Dies funktioniert bis zu einer Vergrößerung von 2.5x. Somit lassen sich ganz kleine Fossilien noch nicht wirklich scharf darstellen.

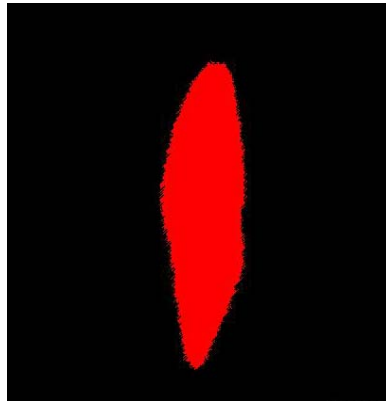


Abb. 27 unscharfe Aufnahme; Fläche = 20381 Pixel

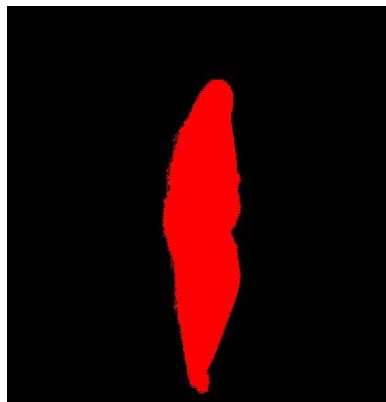


Abb. 28 scharfe Aufnahme; Fläche = 18497 Pixel

8.5 Interpolation

Der jetzige Autofokus funktioniert recht gut, aber er braucht sehr lange, da bis zu 50 Schritte ausgeführt werden müssen. So kommt die Idee, dass nur drei Punkte angefahren werden und der schärfste Punkt errechnet wird. Der schärfste Punkt ist der Scheitelpunkt einer Parabel. Die Parabel anhand dreier Punkte lässt sich mit LabView einfach realisieren, da schon ein Funktionsblock existiert, der die Koeffizienten eines Polynoms anhand einer gewissen Anzahl Punkten berechnet. Da es sich in unserem Fall um eine Parabel handelt, wird ein Polynom zweiter Ordnung erzeugt. Bei den Tests liefert diese Methode jedoch überhaupt keine verlässlichen Resultate, da die Standardabweichung aufgrund des Bildrauschens konstant ein

bisschen schwankt und somit die Parabel recht ungenau wird. Auch der Mittelwert über fünf Messungen und die höhere Ordnung des Polynoms in Verbindung mit mehr Messpunkten bringt keine wirklich besseren Resultate. Somit ist diese Methode als Studie recht interessant, aber für die wirkliche Anwendung überhaupt nicht geeignet.

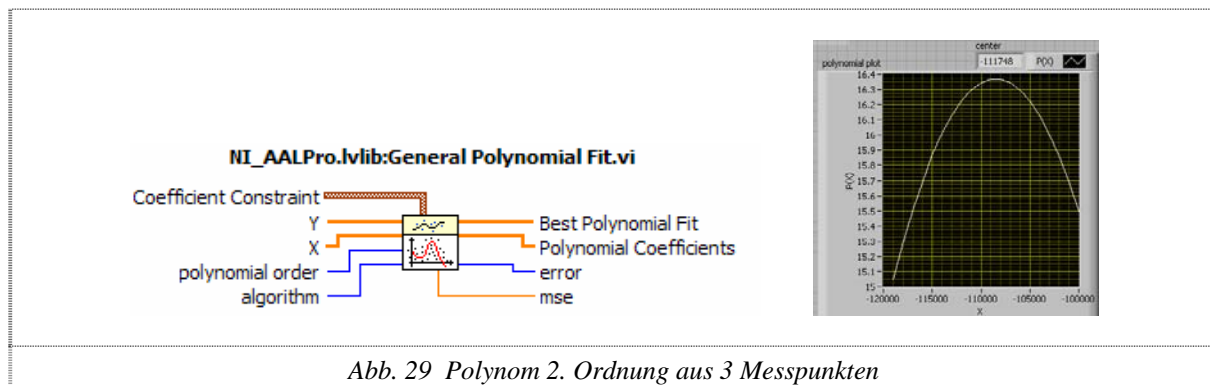


Abb. 29 Polynom 2. Ordnung aus 3 Messpunkten

8.6 ROI (Region of Interest)

Die Kombination mit dem Histogramm und der minimalen Fläche benötigt zum Teil 20 Sekunden für das Fokussieren. Somit suchen wir eine Variante, wie schneller fokussiert werden kann. Dazu darf der Fokus anhand des Histogramms nicht mehr auf den dominanten Hintergrund fallen. Ein ROI (Region of Interest) mit dem Fossil in der Mitte soll das Problem beheben, da der Hintergrund so nur noch einen kleinen Teil des Bildes ausmacht und somit auch keinen grossen Einfluss mehr hat. Dazu wird per Partikelanalyse das kleinste umschliessende Rechteck bestimmt und dieses als ROI verwendet. Die Ergebnisse sind besser als erwartet und auch bei einer Auflösung grösser als 2.5x wird noch zuverlässig fokussiert. Jetzt hat sogar die Lichtquelle keinen grossen Einfluss mehr auf die Fokussierung. Es werden zwar immer noch etwa 10 Sekunden benötigt, aber dafür ist jetzt das Bild unter allen Bedingungen wirklich scharf.

In LabView selber wenden wir nicht direkt die Funktion ROI an, sondern schneiden das umschliessende Rechteck des Fossils aus. Das kleinste umschliessende Rechteck bekommen wir aus der Partikelanalyse mit der Funktion *RoundingRect*.

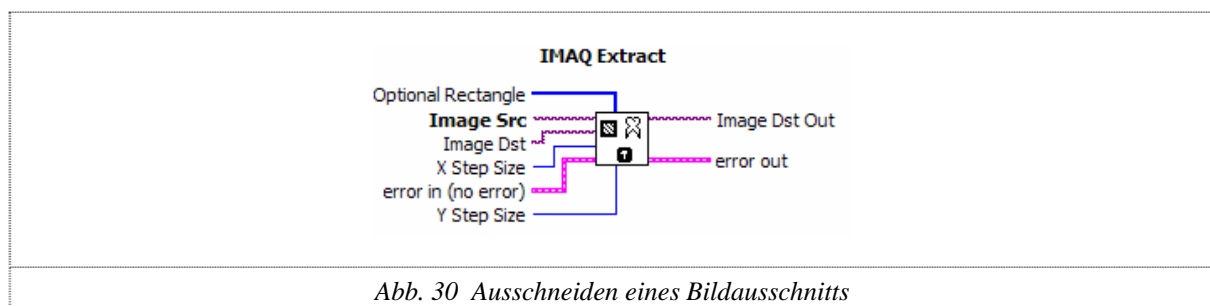


Abb. 30 Ausschneiden eines Bildausschnitts

8.7 Verzögerung

Es wird nach Gründen gesucht, warum der Motorfokus so träge reagiert und somit verantwortlich für die langsame Fokussierung ist. Nach genauem Untersuchen des *serial.vi* haben wir festgestellt, dass mit einer Verzögerung von 500 Millisekunden auf die COM-Schnittstelle geschrieben wird. Nach dem Herabsetzen der Verzögerung auf eine Millisekunde braucht der Autofokus noch etwa 4 Sekunden. Ein totales Weglassen dieser Verzögerung verursacht manchmal Probleme bei der Initialisierung, so dass die ganze Software blockiert wird.

9 Automatisches Ausrichten

Das *Autoalignment* ist für die automatische Ausrichtung der Fossilien zuständig. Dies ist nötig, da die einzelnen Fossilien von Hand auf den Slide aufgeklebt werden, und diese somit nicht ganz vertikal und zentriert stehen. Wir unterscheiden zwischen dem automatischen Zentrieren, Neigen und Rotieren (Drehen), da die Algorithmen grundlegend verschieden sind. Die Korrekturwerte für die Zentrierung und Rotation lassen sich aus einem Bild berechnen, aber die Ausrichtung der Neigungsachsen muss mittels Ausprobieren und langsamem Annähern bestimmt werden.

9.1 Autocenter

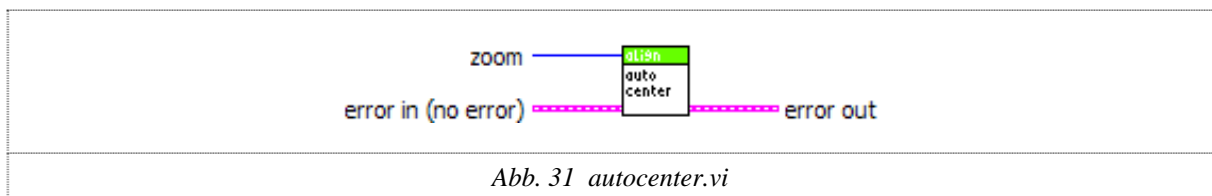


Abb. 31 autocenter.vi

Für das Rotieren oder Vergrössern des Fossils muss dieses in der Mitte des Bildes sein. Da aber die einzelnen Fossilien nicht immer exakt in der Mitte der Felder platziert sind, ist eine automatische Zentrierung notwendig. Mit der Partikelanalyse wird das kleinste umschliessende Rechteck gebildet. Aus seinen Koordinaten lässt sich jetzt die Abweichung von der Bildmitte berechnen. Für die Korrektur muss die aktuelle Vergrößerung des Mikroskops bekannt sein, da die Anzahl zu korrigierender Schritte des Schwenktisches von der Vergrößerungsstufe des Mikroskops abhängig ist.

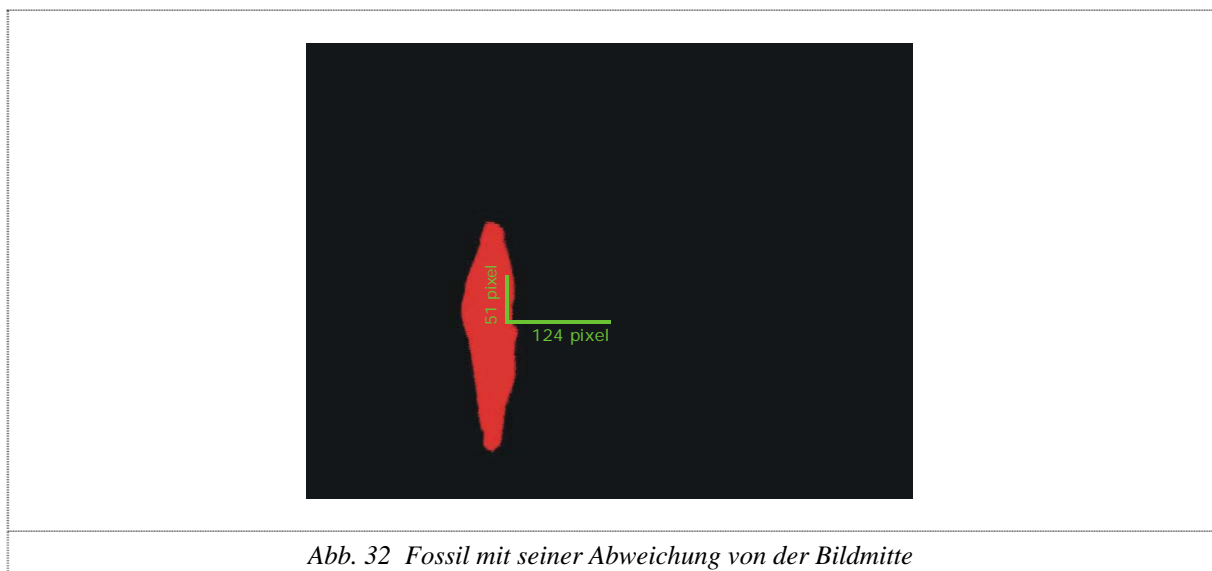


Abb. 32 Fossil mit seiner Abweichung von der Bildmitte

9.2 Autotilt

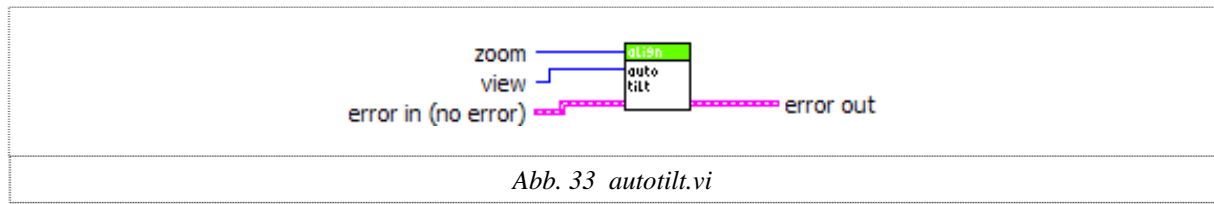


Abb. 33 autotilt.vi

Im Gegensatz zum Autocenter kann hier die beste Position nicht rechnerisch bestimmt werden. Autotilt besteht aus zwei Teilen: dem automatischen Ausrichten der Pitch-Achse und der Roll-Achse. Für die Ausrichtung wird ein ähnlicher Algorithmus wie beim Autofokus verwendet. Bei der Roll-Achse wird je nach Lage der Objekte auf minimale oder maximale Fläche ausgerichtet. Fossilien mit Kielansicht werden auf die minimale Fläche optimiert, Spiral- oder Umbilikalansicht verlangt die maximale Fläche als Parameter. Die Pitch-Achse hat als Richtkriterium die maximale Objektlänge. Anfangs versuchen wir beim Autopitch auf die grösste Fläche zu fahren. Dies funktioniert jedoch nicht zuverlässig, da mit dieser Methode die Fläche kaum ändert. Ein genaueres Resultat ergibt sich, wenn stattdessen die Distanz als Parameter verwendet wird.

autotilt.vi verwendet *autocenter.vi*, um nach jedem Durchgang das Objekt wieder in die Mitte zu schieben und so zu verhindern, dass das Objekt aus dem Bild verschwindet.

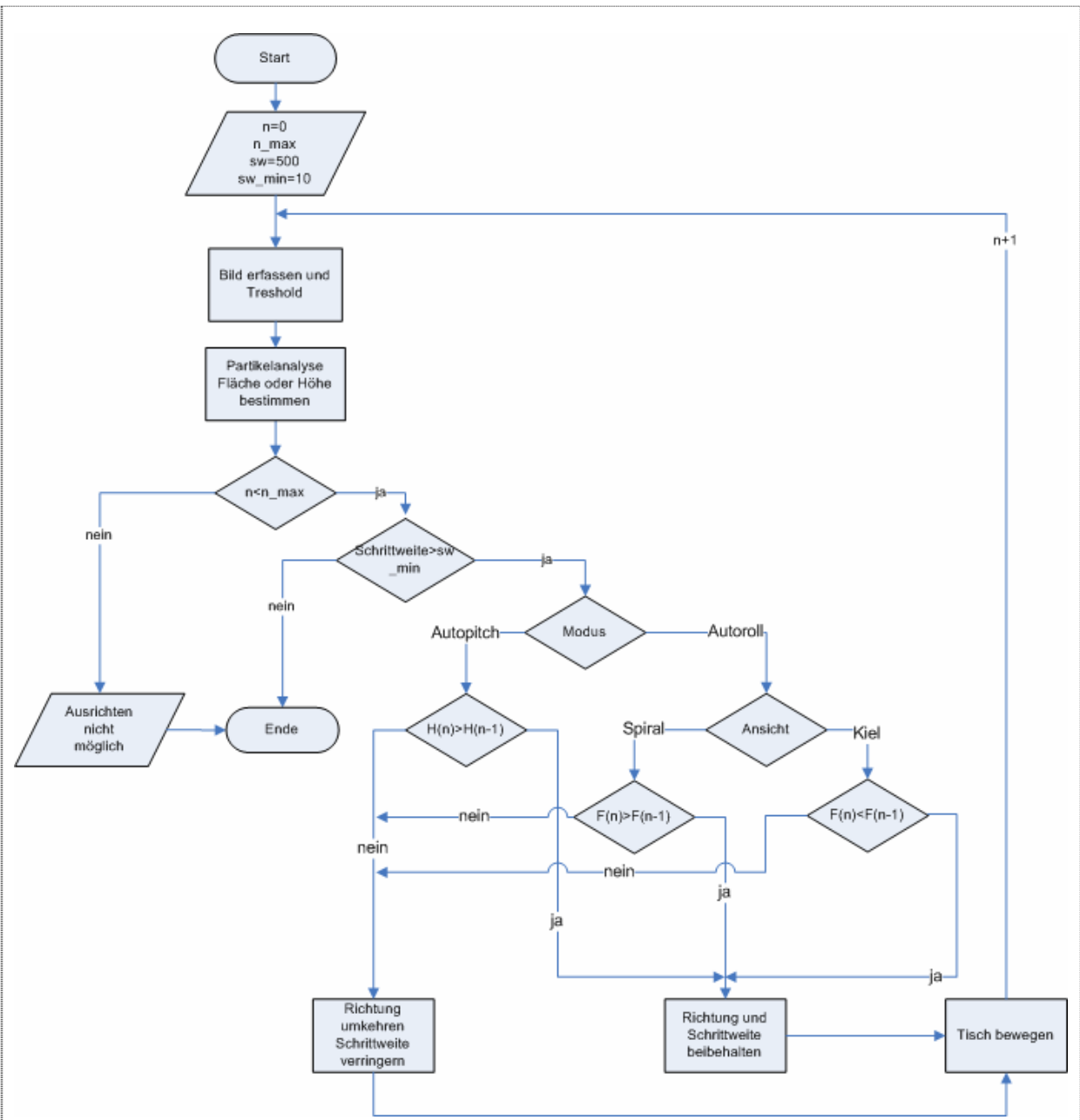
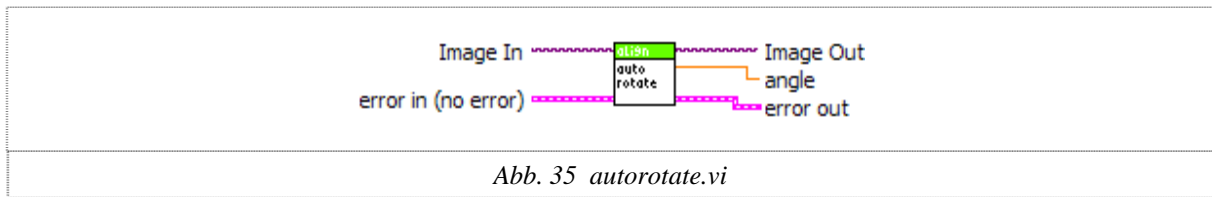


Abb. 34 Ablaufdiagramm Autotilt

9.3 Autorotate



Da der Schwenktisch keine Rotationsachse besitzt, muss dies von der Bildverarbeitung erledigt werden. Wird jedoch ein Bild gedreht, so geht Information verloren und neue Ausschnitte kommen dazu. Aus diesem Grund wird dieser Schritt nur zum Anzeigen und am Schluss zum Abspeichern ausgeführt.

Für das Bestimmen der Lage des Objekts kommt die Methode *Momentum of Inertia* zum Zug. Dies entspricht dem Flächenträgheitsmoment in der Mechanik, mit dem Unterschied, dass an Stelle des Massenelementes der Grauwert an einem Pixel des Bildes oder des Objektes verwendet wird.

Das Flächenträgheitsmoment ist dann $I = \sum [G_i * R_i^2]$,
 G_i ist der Grauwert auf Pixel P_i mit Koordinaten (X_i, Y_i) , und R_i ist der Vektor vom Drehpunkt nach P_i .
Dabei ist $(R_i)^2 = (X_i)^2 + (Y_i)^2$.

Die Methode berechnet also I über das Objekt und findet die optimale Ausrichtung, wenn I minimal wird. Die Methode muss demnach folgende Voraussetzungen erfüllen:

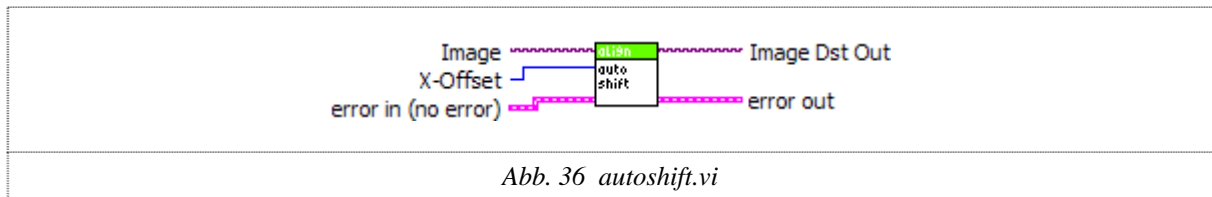
Die Schalen müssen vorausgerichtet sein, d.h. schon einigermaßen vertikal im Slide aufgestellt sein. Wenn die Schalen z.B. mit der Kiel-Achse eher horizontal liegen, dann wird diese Methode die Schalen in diese Richtung ausrichten. Der Grund ist, dass das Flächenträgheitsmoment eine Funktion des Drehwinkels ($= \varphi$) ist. Mit φ ist der Winkel zwischen der Längsachse der Schale in der Kielansicht und der X-Achse des Bildes gemeint.

Die Funktion $I(\varphi)$ ist periodisch, sie wird minimal bei $\varphi = 0$ Grad (Längsachse der Schale ist vertikal), $\varphi = 90$ Grad (Längsachse der Schale ist horizontal), $\varphi = 180^\circ$, etc. $I(\varphi)$ wird maximal, wenn die Schale diagonal im Feld steht (d.h. bei $\varphi = 45$ Grad, 135 Grad, etc.).

Die Bedingungen für die Vertikalausrichtung der Schale über das Flächenträgheitsmoment ist also: Die Schalen müssen vororientiert sein, so dass $45^\circ < \varphi < 135^\circ$. Das ist in unserem Falle zutreffend.

Die Funktion *Momentum of Inertia* wird vom Funktionsblock Partikelanalyse unterstützt und wird als Orientation bezeichnet. Dieser liefert die Abweichung von der Vertikalen in Grad und übergibt sie dem Block *Rotate*, der die Rotation ausführt. Die jetzt neu dazugekommenen Flächen an den Ecken werden schwarz aufgefüllt.

9.4 Autoshift



Da die gemachten Aufnahmen weiter für Bildauswertungen verwendet werden, muss das Objekt an einem genau vordefinierten Ort platziert sein. Diese Funktion wird von *autoshift.vi* übernommen. Das Fossil wird auf die Y-Achse zentriert und die linke Kante vom Objekt auf einen vordefinierten Abstand vom Rand ausgerichtet. Im Gegensatz zu *autocenter.vi*, funktioniert dieses VI ohne den Tisch bewegen zu müssen. Es berechnet die Lage der linken Kante und dem Mittelpunkt der Höhe. Daraus lässt sich die auszuführende Verschiebung berechnen. Die Verschiebung erfolgt rein mit Bildverarbeitung, indem das Bild verschoben und der leere Teil mit schwarz aufgefüllt wird. Der Grund, dass diese Verschiebung nur softwaretechnisch erfolgt ist folgender: Nach dem Rotieren stimmen die reellen X- und Y-Achsen des Tisches nicht mehr unbedingt mit denen des Bildes überein.

Das Verschieben kann aber vom technischen Aspekt her erst nach allen Ausrichtungen (Tilt & Rotate) erfolgen, da diese die Objektlage verändern. Somit erfolgt diese Verschiebung erst unmittelbar vor dem Abspeichern des Bildes.

10 Software

Das verwendete LabView 8.20 verfügt neu, wie nahezu alle anderen Programmiersprachen auch, über einen Projektexplorer. In ihm lassen sich ganze Verzeichnisbäume erstellen, welche für grosse Übersicht in der Projektverwaltung sorgen. Zudem werden Compiler und Installer-Umgebung direkt hier gestartet.

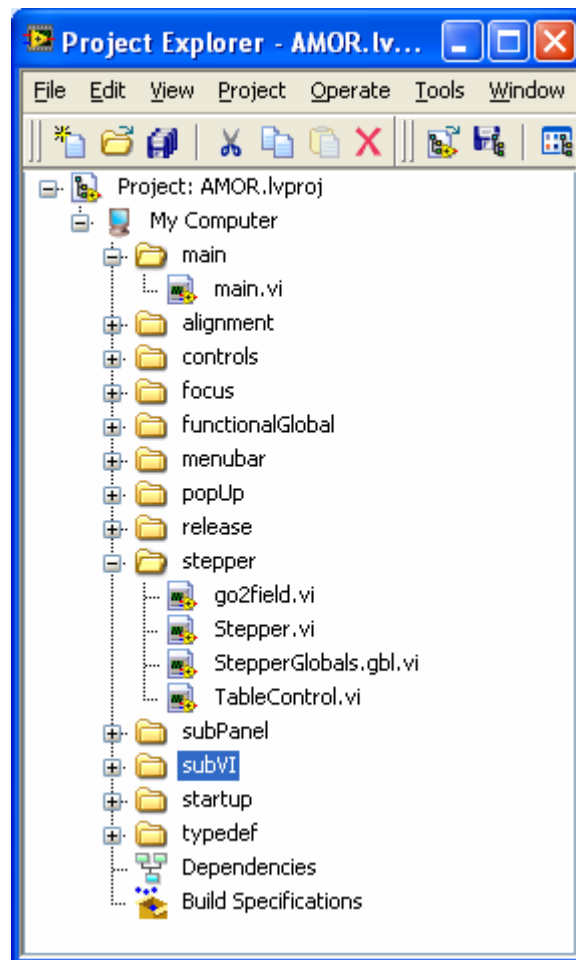


Abb. 37 Projektexplorer

10.1 Projektgliederung

main

Im Ordner main liegt lediglich *main.vi*

main.vi:



main.vi ist das Herz der Software. Es ist als State-Machine aufgebaut und kontrolliert Aufstarten, Herunterfahren, Standby-Zustand und die Parametererfassung vor den Messungen.

alignment

Unter der Rubrik *alignment* sind alle VI's untergebracht, welche sich mit der Ausrichtung des Fossils beschäftigen.

autocenter.vi:



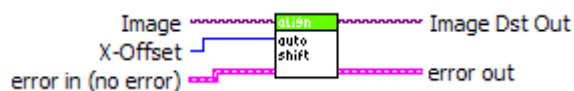
autocenter.vi schiebt ein gefundenes Mikrofossil in die Mitte des Kamerabildes (durch Bewegen der Tischachsen).

autorotate.vi:



autorotate.vi rotiert ein gefundenes Fossil, um es gerade zu stellen. Die Winkelabweichung wird mittels *momentum of inertia* erkannt. *autorotate.vi* funktioniert nur bei Fossilien in Kielansicht!

autoshift.vi:



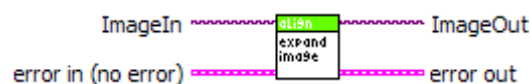
autoshift.vi verschiebt das Fossil so, dass es vom linken Bildrand einen konstanten Abstand (X-Offset) hat und in Y-Richtung zentriert ist. *autoshift.vi* funktioniert rein bildtechnisch, es nimmt also keine Achsensteuerung am Schwenktisch vor.

autotilt.vi:



autotilt.vi neigt die Fossilien in die richtige Lage (ansteuern der Pitch- und Roll-Achse). *autotilt.vi* verwendet dazu einen approximativen Algorithmus.

expandImage.vi:



expandImage.vi bringt das Bild in das NTSC-Format, also 640x480 Bildpunkte. Dies ist notwendig, weil die Sony-Kamera nur ein gutes 630x480-Pixel-Bild liefert.

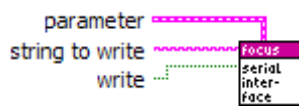
controls

Hier sind die selbst erstellten Buttons ("Pfeilbuttons") untergebracht.

focus

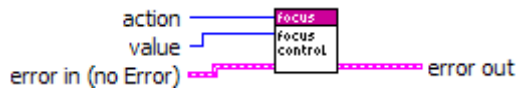
Dieses Verzeichnis beinhaltet die serielle Kommunikation, den Fokuszugriff sowie den Autofokus.

serial.vi:



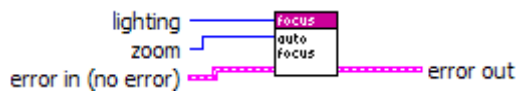
serial.vi erlaubt es, einen String über die serielle Schnittstelle zu senden.

FocusControl.vi:



FocusControl.vi steuert den Motorfokus an. Dieses VI kann den Motorfokus initialisieren und ermöglicht absolutes oder relatives Bewegen des Mikroskopkörpers.

autofocus.vi:



autofocus.vi fokussiert das Bild mittels eines approximativen Algorithmus. Gestartet wird an der Standardfokusposition, welche von *fGbDefaultFocusPosition.vi* zur Verfügung stellt.

functionalGlobal

Hier sind alle *functional Global Variablen* untergebracht. (Erklärung zu *functional Global Variablen* in Kapitel 10.6).

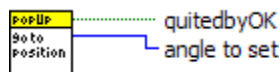
menubar

Speicherort der Runtime-Menubar.

popUp

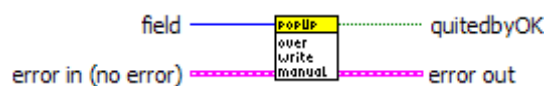
Hier sind alle PopUp-Fenster untergebracht, wenn sie nicht der Logik nach in einen anderen Ordner gehören (z.B. *autofocus.vi*). Dies sind im Wesentlichen das Errorhandling, die Messpositionswahl, Einstellungs- und Speicherdialoge.

gotoPosition.vi:



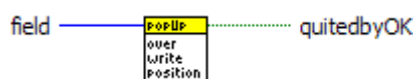
gotoPosition.vi bewegt alle Achsen in die vorgegebenen Achsenpositionen. Da der Rotationswinkel nicht angesteuert, sondern nur bildtechnisch gedreht wird, liefert dieses VI diesen Wert zurück.

overwriteManual.vi:



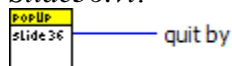
overwriteManual.vi kommt in der Automatikmessung zum Einsatz. Es lässt den Anwender eine Messposition manuell überschreiben.

overwritePosition.vi:



overwritePosition.vi wird ebenfalls in der Automatikmessung verwendet. Es lässt dem Bediener die Möglichkeit, eine nicht gefundene Messposition manuell zu setzen.

Slide36.vi:



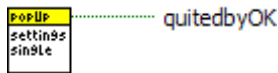
In *Slide36.vi* lassen sich Messpositionen abschalten.

Slide60.vi:



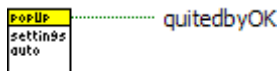
In *Slide60.vi* lassen sich Messpositionen abschalten.

settingsSingle.vi:



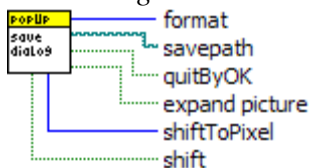
settingsSingle.vi ist die Eingabemaske der Parameter für die Einzelmessung.

settingsAuto.vi:



settingsAuto.vi ist die Eingabemaske der Parameter für die Automatikmessung.

saveDialog.vi:



saveDialog.vi ist der Speicherdialog in der Einzelmessung.

release

In *release* sind die Benutzeraufforderungen während des Ausschaltvorganges untergebracht. Diese VI's haben technisch keine Funktion, sie stoppen die Software nur bis der Anwender bestätigt hat, dass das Gerät ausgeschaltet ist.

r_lightsource.vi:



r_lightsource.vi fordert den Bediener auf, die Lichtquelle auszuschalten.

r_microscope.vi:



In *r_microscope.vi* wird der Anwender dazu angehalten, den Motorfokus auszuschalten.

r_stepperbox.vi:

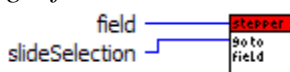


Der Benutzer wird aufgefordert, die Schrittmotorbox auszuschalten.

stepper

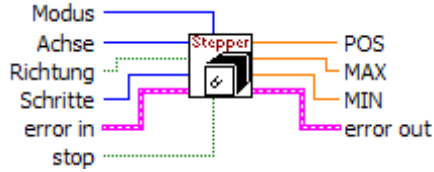
Hier ist die gesamte Schwenktischsteuerung zu finden.

go2field.vi:



go2field.vi setzt eine Feldnummer (je nach Slide) in ihre Koordinaten um und bewegt den Schwenktisch an diese Position.

Stepper.vi:

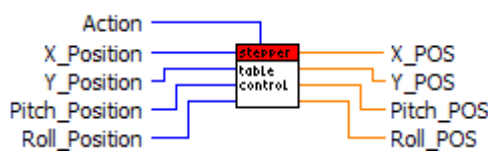


Stepper.vi ist der Motorentreiber zum Schwenktisch, welcher von Herrn Daniel Binggeli der Fachhochschule Nordwestschweiz Brugg entwickelt wurde. An ihm werden minimale Veränderungen vorgenommen. Technisch hat dies keine grosse Bedeutung, da lediglich die Achsen durch Typdefinitionen ersetzt werden, um zu verhindern, dass ein Anwender eine Achse auswählt, die das VI nicht kennt.

StepperGlobals.gbl.vi:

StepperGlobals.gbl.vi sind die globalen Variablen (Achsenpositionen) von *Stepper.vi*. Es handelt sich ebenfalls um einen Teil von Herrn Daniel Binggelis Arbeit.

TableControl.vi:



TableControl.vi übernimmt die Ansteuerung des Schwenktisches. Dieses VI kann den Tisch initialisieren, Achsen absolut oder relativ bewegen und die Achspositionen zurückgeben. Das Ansprechen des Schwenktisches geschieht immer über dieses VI.

subPanel

Alle Subpanels, die im Panel von *main.vi* während des Betriebes geladen und ausgeführt werden, sind hier zu finden.

spnlAdjust.vi:



spnlAdjust.vi unterstützt den Anwender beim Justieren des Mikroskops und bei der Beleuchtungseinstellung.

spnlAuto.vi:



In *spnlAuto.vi* ist die gesamte Automatikmessung untergebracht.

spnlBlank.vi:



spnlBlank.vi ist ein VI ohne jegliche Funktion und Anzeige. Es ist nur ein leeres Panel.

spnlDefaultFocus.vi:



Erlaubt dem User den Fokusstartwert zu setzen. Je näher der Startwert beim optimalen Endfall ist, desto schneller wird automatisch fokussiert.

spnlInit.vi:



spnlInit.vi ist der Anzeigebildschirm beim Aufstarten.

spnlRelease.vi:



spnlRelease.vi wird beim Herunterfahren der Software angezeigt.

spnlSingle.vi:



spnlSingle.vi behandelt den gesamten Einzelmessungsmodus.

spnlStandby.vi:



spnlStandby.vi ist der Wartezustand. Es kann zwischen Einzel-, Automatikmessung oder Beenden des Programms gewählt werden.

subVI

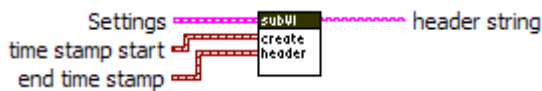
Speicherorte der VI's, die Berechnungsaufgaben übernehmen, Standarde liefern, Daten zur Verfügung stellen usw. Im Prinzip SubVI's nach klassischem LabView-Prinzip.

addEnding.vi:



addEnding.vi fügt dem Filename die Dateiondung hinzu.

createHeader.vi:



In *createHeader.vi* wird der gesamte Header für das Logfile geschrieben.

createSamplePositionString.vi:



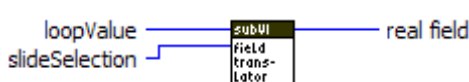
Dieses SubVI kreiert aus den Positionsdaten der Messung einen String, der ins Logfile geschrieben werden kann.

defaultFocusPosition.vi:



Da ein optimal scharfer Fokus abhängig von der Zoomstufe ist, wird der Variabelwert des Standardfokus zuerst mit der selektierten Zoomstufe verrechnet. Dies geschieht in *defaultFocusPosition.vi*.

fieldtranslator.vi:



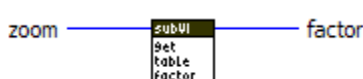
fieldtranslator.vi übernimmt die Umsetzung des reinen Schleifenwertes bei der Automatikmessung in das effektiv gemessene Feld (→ Slide wird in Mäanderform abgefahren, weil dies zeitoptimiert ist).

filenameIsStandard.vi:



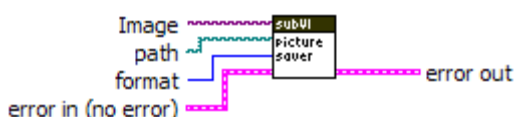
Diese VI überprüft, ob der übergebene String mit einem Standardfilename für eine Messung übereinstimmt.

getTableFactor.vi:



In *getTableFactor.vi* ist die Umrechnung von Pixel zu Anzahl Schritten der Schrittmotoren realisiert. Logischerweise ist dies abhängig von der gewählten Zoomstufe.

pictureSaver.vi:



pictureSaver.vi ist das VI, das die Speicherung eines Bildes übernimmt. Es verlangt als Parameter ein Bild, einen Speicherort (Dateiname inklusive Pfad) und das Speicherformat.

relativePath.vi:



Dieses VI generiert, unabhängig davon ob die Software in einem Standalone-System oder in der Entwicklungsumgebung läuft, eine Referenz auf ein anderes VI, dessen Pfad relativ angegeben werden kann.

startup

In der Rubrik *startup* befinden sich alle Popup-Fenster, die den Benutzer zu Aktionen auffordern (Geräte einschalten, Schalter richtig setzen). Zusätzlich ist die Wahlmaske für den Aufstartvorgang (*startup.vi*) hier abgespeichert.

lightsource.vi:



Der Benutzer wird aufgefordert, die Lichtquelle einzuschalten.

microscope.vi:



microscope.vi hält dazu an, den Motorfokus einzuschalten.

microscopeOffset.vi:



Der Anwender wird dazu angehalten, das Mikroskop in die richtige Stellung zu bringen.

selectors.vi:



Der Bediener wird aufgefordert, die Schalter am Mikroskopkörper korrekt zu setzen.

stepperbox.vi:



stepperbox.vi ist die Anfrage, ob die Schrittmotorbox eingeschaltet ist.

startup.vi:



startup.vi ist die Maske, welche vom Anwender wissen will, wie er das System hochfahren will (einfach oder geführt) und ob ein Einstellen der Polfilter gewünscht wird.

typedef

Hier befinden sich alle Typedefinition des ganzen Projekts (Erklärung zu Typedefinitionen in Kapitel 10.7).

10.2 main

main.vi ist das Herz der gesamten Software. Es ist als State Machine realisiert, was für eine einfache Handhabung sorgt. Somit ist es jederzeit möglich, an jegliche Programmstelle zu springen. Dies ist bei Steuerprogrammen dieser Art oft von Vorteil.

main.vi übernimmt das Aufstarten des gesamten Aufbaus, behandelt das Vorbereiten von Messungen und das Freigeben der Geräte beim Ausschalten. Für die effektiven Messalgorithmen (Einzel- und Automatikmessung) wird die Kontrolle an SubVI's übergeben, beim Beenden dieser Algorithmen wechselt sie wieder zurück zu *main.vi*.

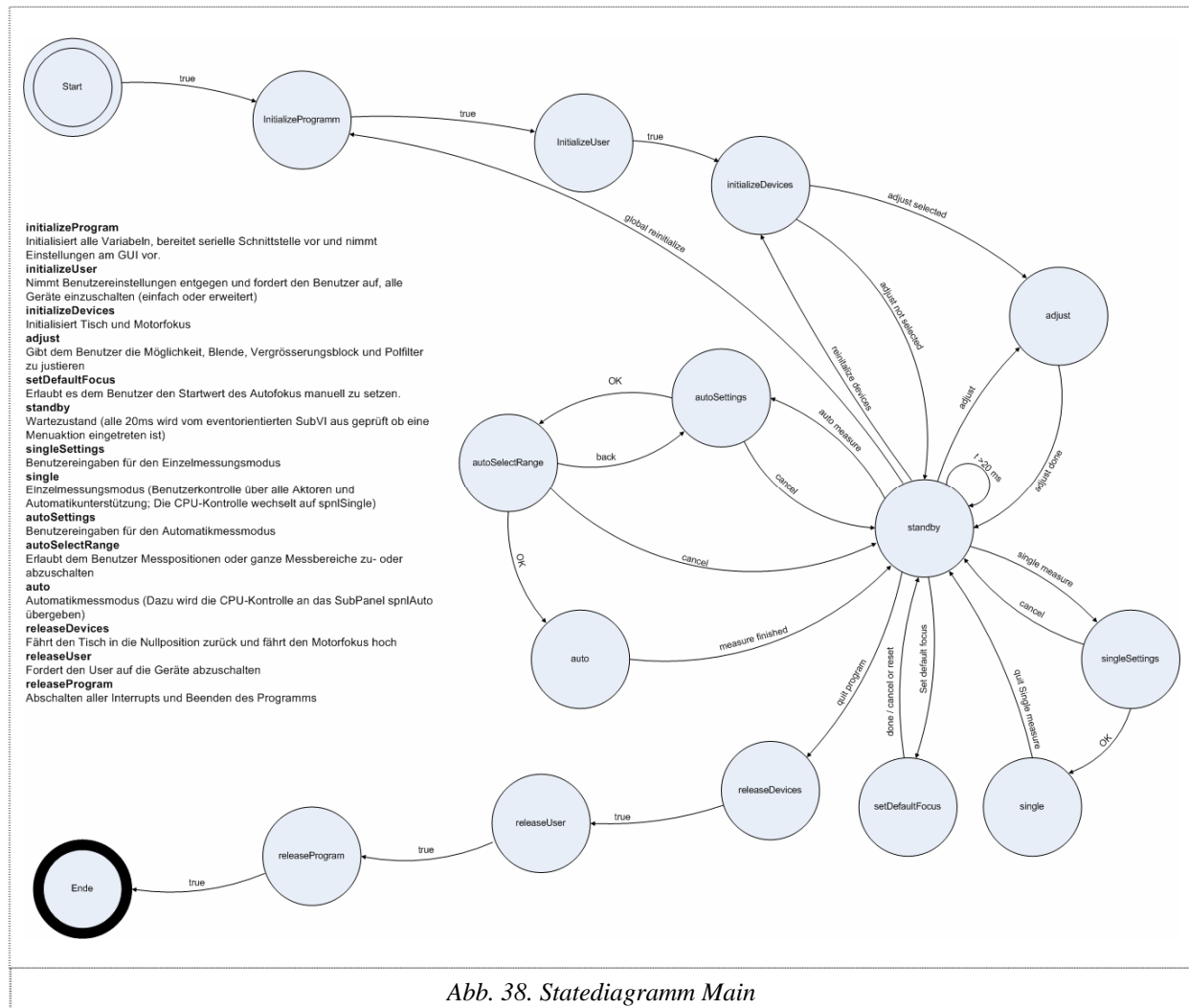


Abb. 38. Statediagramm Main

Aufstarten

Die Initialisierung läuft grundsätzlich in drei Schritten ab. Zuerst wird die ganze computerseitige Initialisierung (*initializeProgram*) vorgenommen. Hier werden zum Beispiel Variablen initialisiert, die serielle Schnittstelle vorbereitet oder das Anzeigebild geladen. In einer zweiten Phase (*initializeUser*) wird der User aufgefordert, gewisse Parameter anzugeben und alle Geräte einzuschalten. Im letzten Initialisierungsschritt (*initializeDevices*) werden alle externen Geräte (Tisch und Mikroskop) aufgestartet. Unter Umständen folgt jetzt noch das unterstützte Justieren gewisser Mikroskopteile. Sonst wird direkt auf *standby* geschaltet.

Standby

standby lässt das Programm auf dem Startbildschirm verweilen. Hier wird CPU-sparend gearbeitet. Die Benutzereingaben werden eventorientiert entgegengenommen und die Menubar wird alle 20 Millisekunden überprüft.

Startet der Benutzer eine Messung, so werden zuerst gewisse Messparameter gefordert. Dann wird die Kontrolle an die Messalgorithmen abgegeben.

Release

Wie das Aufstarten geschieht auch das Herunterfahren in drei Schritten. Nur laufen diese hier in umgekehrter Reihenfolge ab. Zuerst werden Tisch und Mikroskop in eine vernünftige Ausgangslage (Tisch eben und zentriert; Mikroskop etwas vom Tisch weg) gebracht. Dann folgt die Aufforderung an den User alle Geräte abzuschalten und zuletzt wird LabView korrekt beendet.

10.3 Automatikmessung (spnAuto.vi)

Wenn alle Parameter gesetzt worden sind, startet der Automatikmodus. Auch hier wird eine State Machine eingesetzt.

Allgemein läuft die Automatikmessung in zwei Schritten ab. Zuerst wird eine Kontrolle (Slidecheck) und dann erst die effektive Messung ausgeführt.

Der Benutzer erhält über den Bildschirm jeweils das zuletzt aufgenommene Bild angezeigt und kann somit, wenn er will, die Messung nachvollziehen.

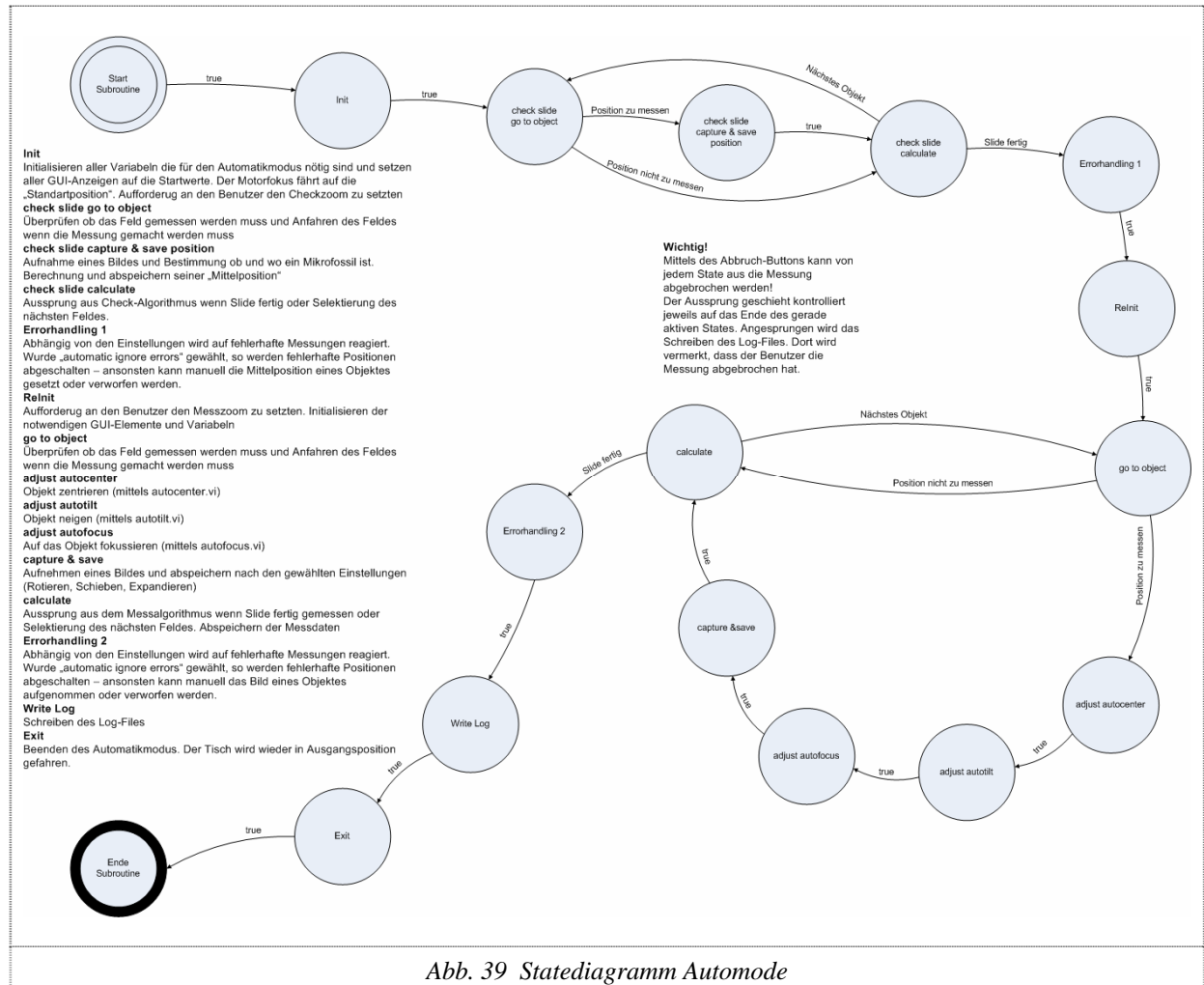


Abb. 39 Statediagramm Automode

Slidecheck

Da nicht alle Fossilien schön in der Mitte des Messfeldes liegen, passiert es oft (vor allem bei grossen Zoomstufen), dass keine Fossilien detektiert wurden. Um dem entgegenzuwirken, wird zuerst eine Positionsüberprüfung gemacht. Der ganze Messstreifen wird vor Beginn der eigentlichen Messung einmal komplett abgefahren und von jedem Fossil die Positionsdaten gespeichert. Zugleich werden „Leerfelder“ eliminiert.

Am Ende der Überprüfung wird je nach gewählten Parametern direkt weitergefahren oder der Benutzer aufgefordert, zu gewissen Feldern manuelle Angaben zu machen.

Messalgorithmus

Erst jetzt startet der effektive Messalgorithmus. Er fährt die einzelnen zuvor gespeicherten Messpositionen an und startet die Messung. Das bedeutet, es wird zuerst auf das Objekt ausgerichtet (*autocenter*), dann wird die Tischneigung eingestellt (*autotilt*), gefolgt von der automatischen Fokussierung (*autofocus*). Schliesslich wird das Bild geschossen und abgespeichert. Das Rotieren oder Verschieben des Bildes (*autorotate / autoshift*) geschieht auch in dieser Phase. Es bedarf also kein separater State, da dies nur ein softwaretechnisches Rotieren oder Schieben des endgültigen Bildes ist.

Sind alle Positionen abgearbeitet, folgen Errorhandling und Schreiben des Log-Files. Somit ist die Messung abgeschlossen und die CPU-Kontrolle geht zurück an *main.vi*.

Mäanderform beim Positionsabfahren

Da das Abfahren des Slides (vor allem von ganz links nach rechts oder umgekehrt) doch eine gewisse Zeit braucht, wird nach einer Zeitoptimierung gesucht. Gefunden wird sie im Abfahren der Messpositionen. Jede zweite Messreihe wird rückwärts zu den Feldnummern abgefahren. So entfällt das "Zurückfahren" quer über den ganzen Messstreifen.

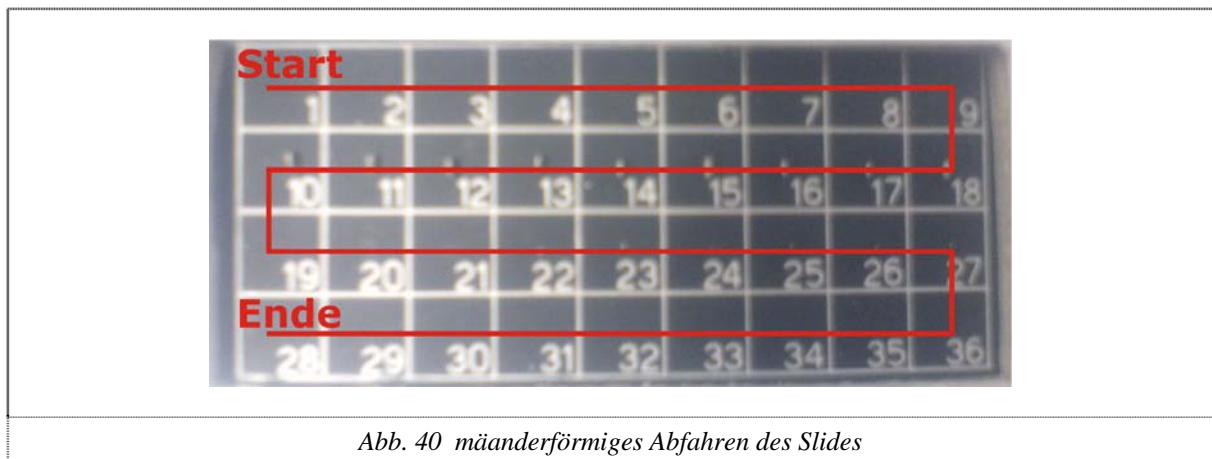


Abb. 40 mäanderförmiges Abfahren des Slides

Logfile

An Ende jeder Automatikmessung wird ein Logfile geschrieben, welches Aufschluss über die gemachte Messung gibt. Es wird am selben Ort abgespeichert, an dem die Bilder liegen und trägt den Dateinamen *log.txt*.

In der Headerinformation des Textfiles stehen die gewählten Parameter sowie Start- und Endzeit der Messung. Für jede Position wird zudem hinterlegt, ob die Messung funktioniert hat. Falls ja, werden alle Aktorendaten gespeichert, was ein späteres Reproduzieren der Messung möglich macht.

10.4 Einzelmessung (*spnlSingle.vi*)

Der Einzelmessmodus lässt dem Benutzer die Möglichkeit, nach seinen eigenen Wünschen ein Bild von einem Mikrofossil zu schiessen und abzuspeichern. Dazu bekommt der Benutzer ein Livebild der Kamera. Die Bedienoberfläche lässt ihn manuell auf alle Aktoren zugreifen und bietet zudem automatische Unterstützung beim Ausrichten und Fokussieren.

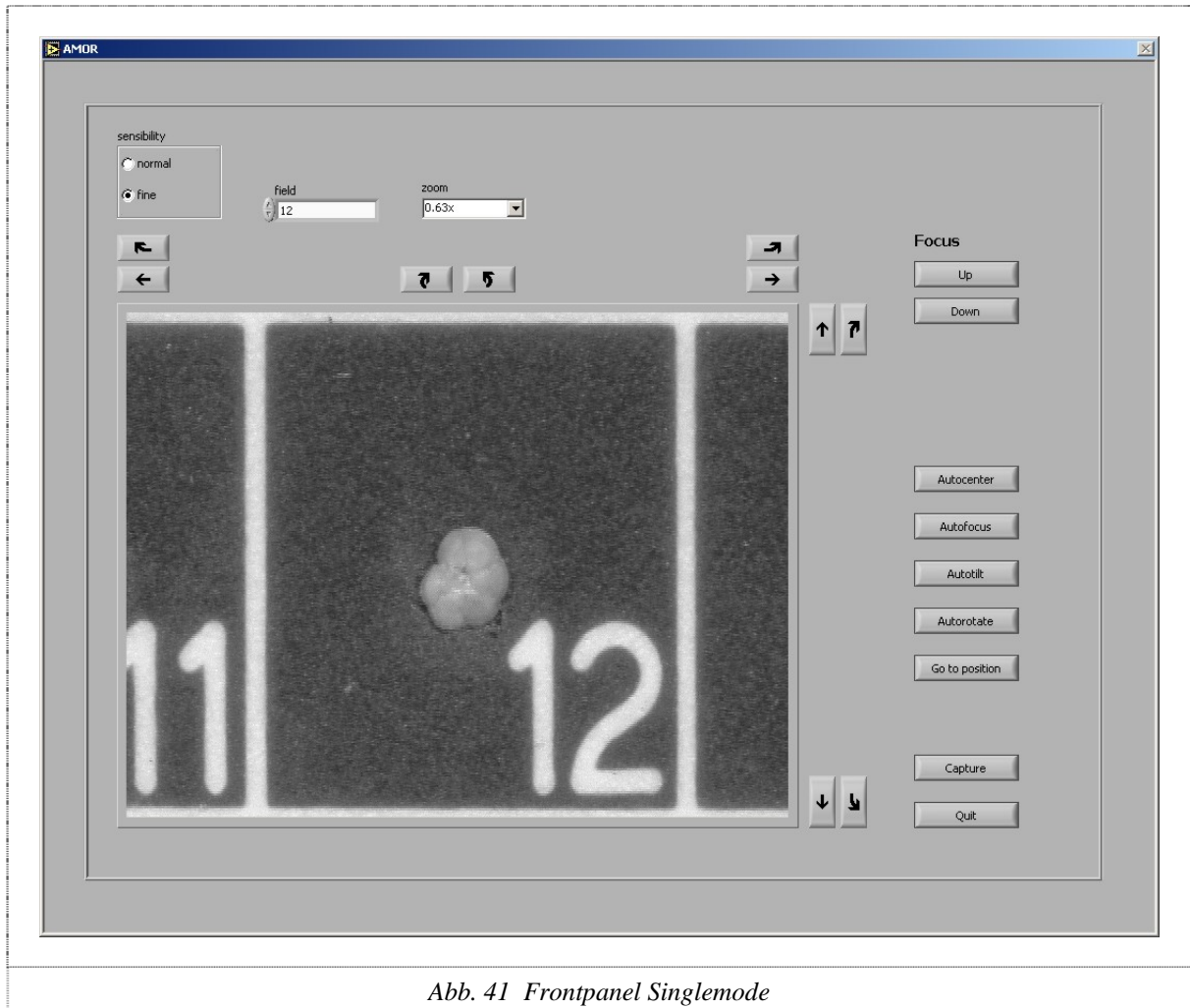


Abb. 41 Frontpanel Singlemode

Die Benutzereingriffe geschehen jeweils direkt in der Eventstruktur, welche die Benutzereingabe erfasst. Die automatischen Funktionen sind als einzelne States realisiert. Sie benötigen den Kamerazugriff und dieser darf nicht mehrfach passieren. Der Sprung auf diese States wird jedoch auch von der Eventstruktur behandelt.

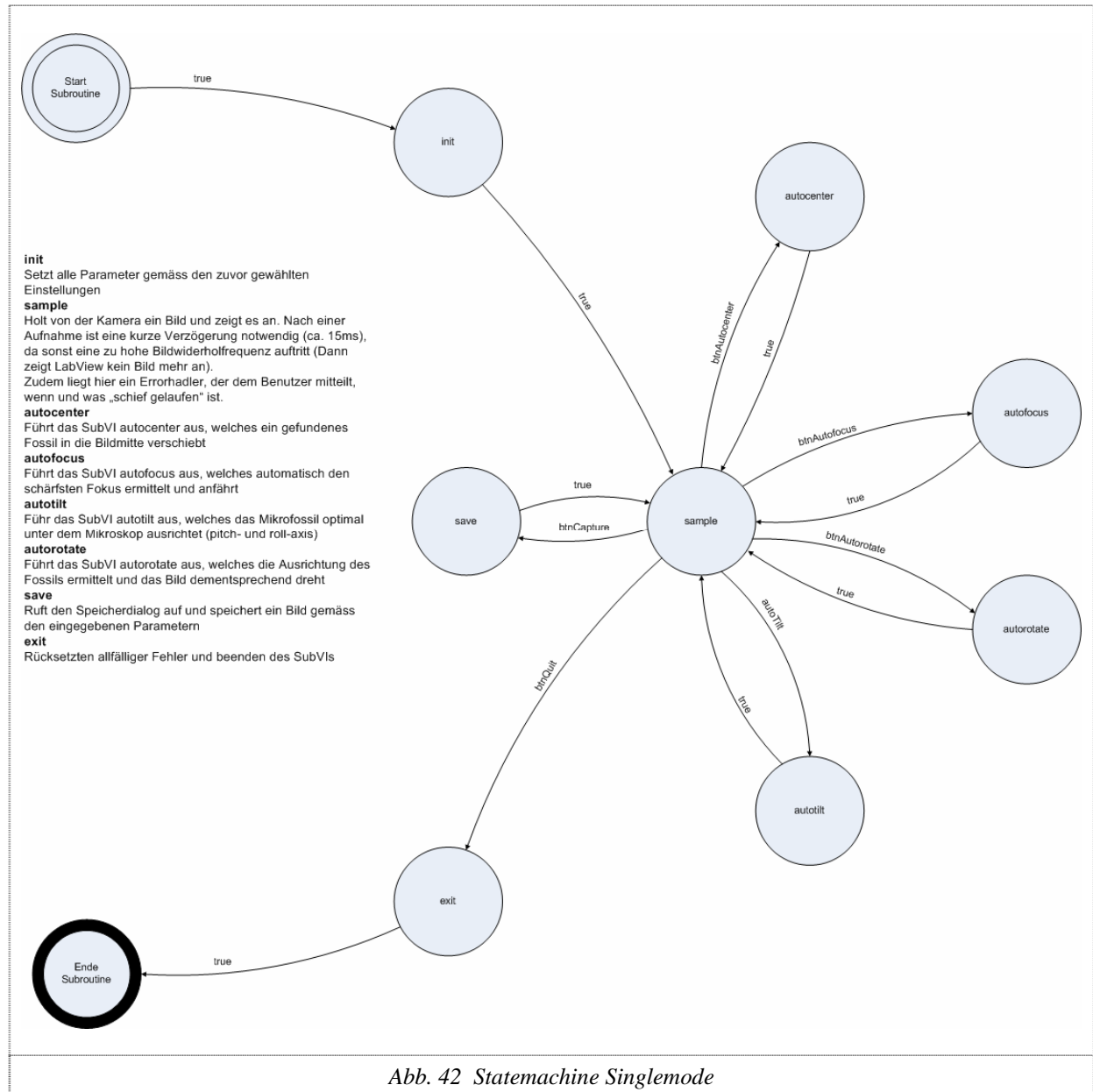


Abb. 42 State machine Singlemode

10.5 Aufbau der Statemachine

Das Hauptprogramm, wie auch alle der komplexeren Unterprogramme, sind als Statemachine realisiert. Dies lässt sich unter LabView sehr einfach erstellen und bietet zugleich eine übersichtliche Struktur.

Der Aufbau umfasst im Wesentlichen 3 Elemente:

- Die while-Schleife mit einem Schieberegister: Sie übernimmt das permanente Wiederholen sowie die Speicherung des nächsten States im Schieberegister.
- Eine Case-Struktur: Sie muss jeden möglichen State beinhalten.
- Ein Enum: Es beinhaltet alle programmtechnisch möglichen States. Am Besten wird es als Typedefinition (typedef) realisiert, denn so können nur States auftreten, welche es auch gibt.

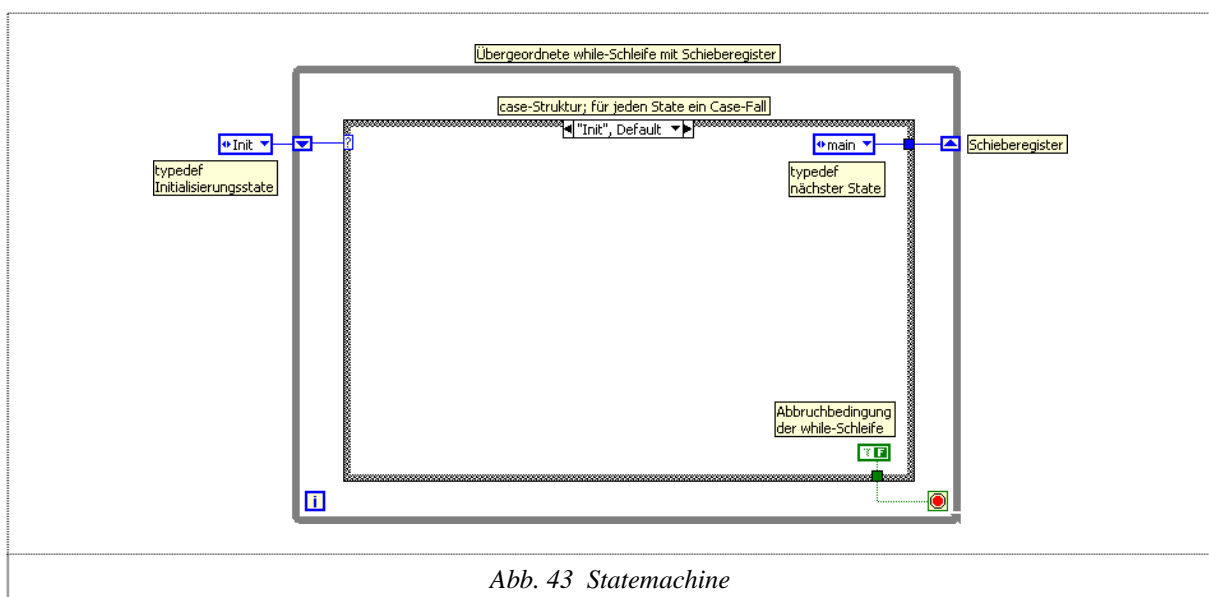


Abb. 43 Statemachine

Funktionsbeschreibung:

Das Schieberegister wird mit einem Initialisierungswert versehen. Dies ist der State, der zuerst aufgerufen wird. Normalerweise wird ein State aufgerufen, der eine Initialisierungsfunktion hat. Die Case-Struktur ruft nun den Case auf, der als Wahlbedingung anliegt (In unserem Fall ist das der State *Init*). Jetzt werden alle Programmteile ausgeführt, welche in dieser Strukturebene liegen. Die Abbruchbedingung hat normalerweise den Wert *false*, denn dann läuft die while-Schleife weiter. Will man die Statemachine beenden, so wird hier *true* gewählt. Ans Schieberegister ist die Typedefinition des nächsten States gehängt. So wird der nächste aufzurufende State im Schieberegister abgelegt. Die while-Schleife prüft jetzt, da die ganze "Ausführebene" abgearbeitet worden ist, ob sie abbrechen kann. Da wir aber ein *false* übergeben haben, ist dies nicht der Fall und die while-Schleife wird erneut aufgerufen. Nun wird der Speicherwert aus dem Schieberegister ausgelesen. Der ist im zweiten Aufruf nicht mehr *Init*, sondern *main*. Jetzt kann die Statemachine um beliebig viele States erweitert werden. Der Entwickler hat nur noch für einen korrekten Aufruf der States zu sorgen.

10.6 Functional Global Variable

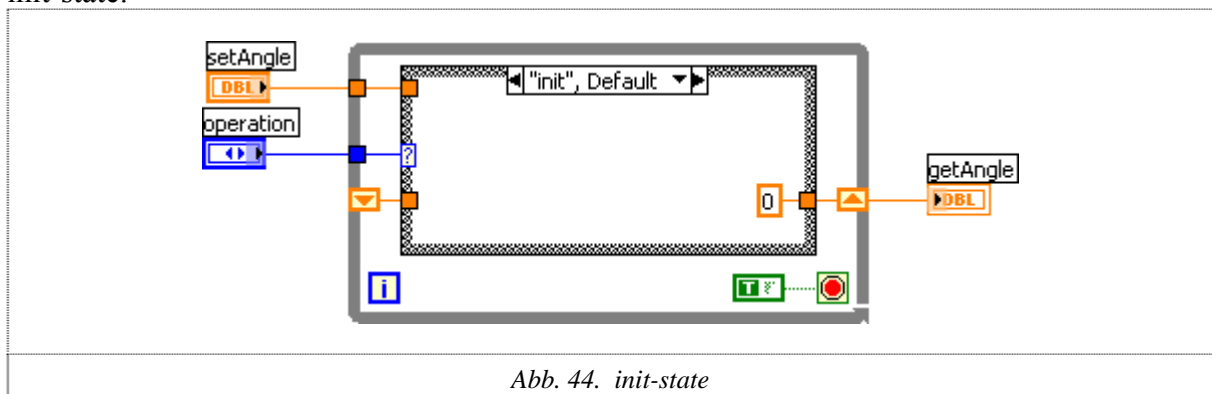
Die Überlegung hinter einer *functional Global Variable* ist ähnlich einem gekapselten Objekt in einer modernen Programmiersprache wie z.B. C++. Die Variabel wird in einem eigenen VI untergebracht, das sie schreiben, lesen oder initialisieren (*init*, *get*, *set*) kann. Somit kann jederzeit von jeder Stelle des Programms gesichert auf den Wert zugegriffen werden.

Funktionsbeschreibung:

Hier werden zwei Feinheiten verwendet, die so nicht direkt von LabView vorgesehen worden sind. Das Nichtinitialisieren eines Schieberegisters und eine *while*-Schleife, die immer abweisend ist. Dank dem Nichtinitialisieren des Schieberegisters ist jeweils der letzte Wert des Schieberegisters aktuell und wird bei Neuaufwurf des VI's (was bei jedem Lesen der Fall ist) nicht überschrieben und hält somit den letzten Wert. Die abweisende *while*-Schleife (die *true*-Konstante, die auf die "loop-condition" geführt ist, bricht die Schleife nach einem Durchlauf ab) sorgt dafür, dass nicht im VI verweilt, sondern sofort zum aufrufenden VI zurückgekehrt wird.

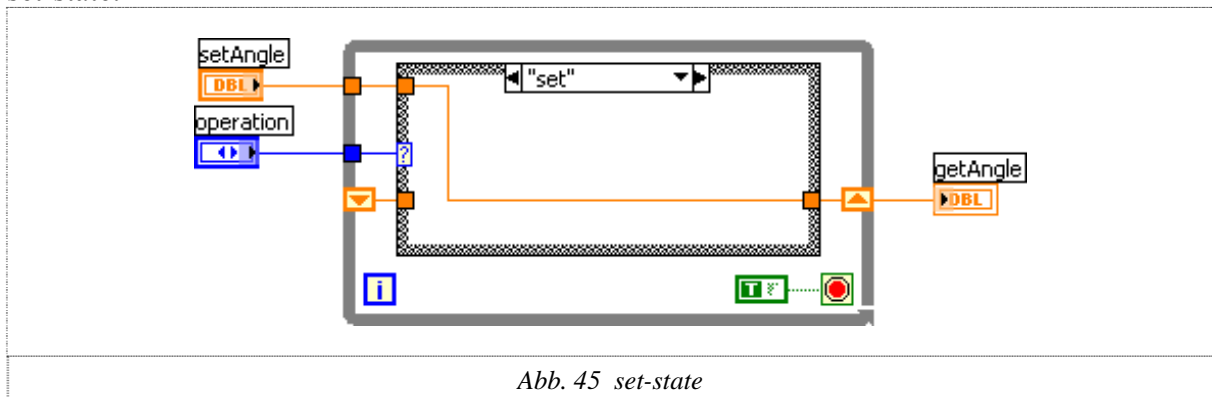
Die *while*-Schleife wird jedoch benötigt, weil nur sie mit einem Schieberegister fungieren kann.

init-state:



Wird als Operation *init* gewählt, so wird der vorgegebene Startwert initialisiert! Ein allfällig übergebener Wert wird nicht beachtet und verworfen.

set-state:



set lässt das Schreiben eines Wertes zu. Der übergebene Wert wird ins Schieberegister geschrieben und somit abgelegt. Theoretisch wird der eben geschriebene Wert wieder

zurückgegeben. Praktisch wird dieser jedoch nicht benötigt, da man ihn sowieso kennt. Er verfällt sozusagen, da der Anschluss ausserhalb des VI's einfach nicht angeschlossen wird.

get-state:

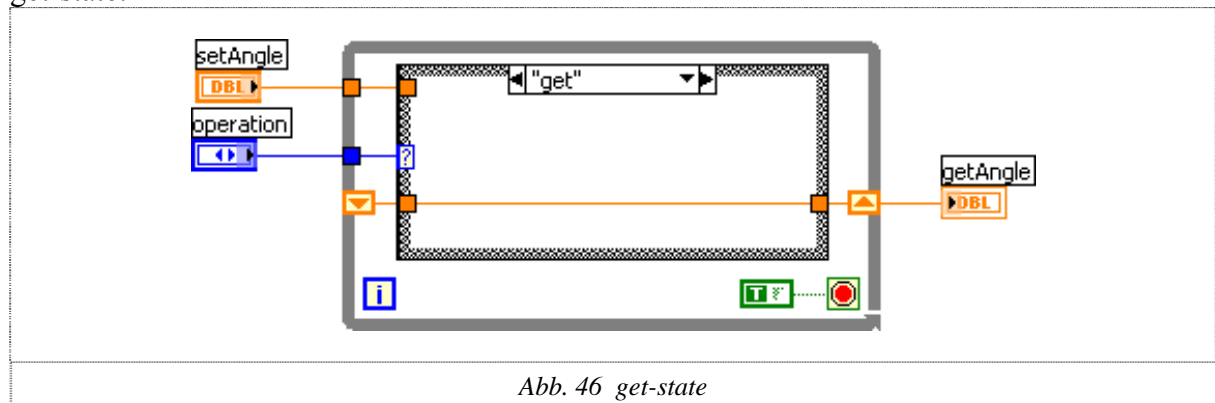


Abb. 46 get-state

Mit *get* kann der Wert der Variable abgerufen werden. Der Speicherwert wird aus dem Schieberegister geholt, dem Übergabecorner übergeben und somit an den Aufrufer zurückgeliefert.

Leicht abgeänderte Varianten

In der Software sind noch zwei leicht abgeänderte Versionen der *functional Global Variable* zu finden. Erstens die Signalisation und die indextierten Speichervariabeln. Ist das Prinzip einer *functional Global Variable* verstanden, so ist es auch nicht schwierig diese beiden Spezialfälle zu begreifen.

Signalisation:

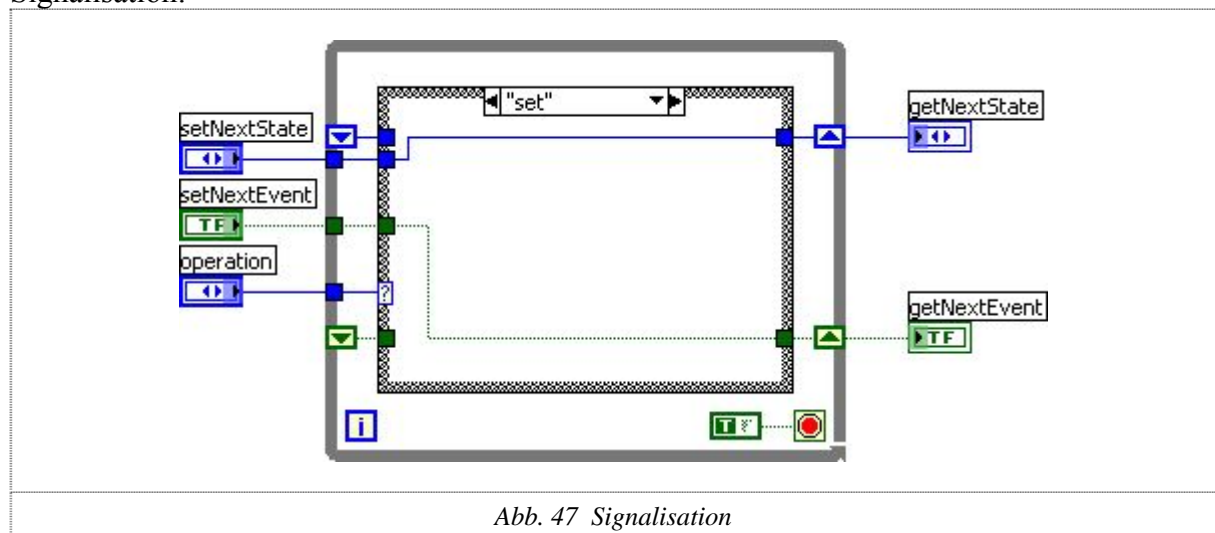


Abb. 47 Signalisation

Es wird hier nur der *set*-Case der Variable gezeigt, da auch *init* und *get* nahezu identisch mit der normalen *functional Global Variable* sind (auf der Dokumentations-CD befindet sich die komplette Softwaredokumentation, auf welchem alle Teilfälle zu sehen sind).

Diese Variable speichert eigentlich zwei Speicherwerte gleichzeitig. Zum Ersten das Signalisationsflag (grüner Signalpfad), welches dazu dient, dem aufrufenden VI anzuzeigen, ob ein Eingriff erfolgt. Zum Zweiten der blaue Signalpfad. Er enthält bei Zugriff auf die Variable einen weiteren Wert, der dem aufrufenden VI zusätzliche Informationen liefert. Im Falle, dass die Variable für die Übersteuerung der Statemachine eingesetzt wird, ist es der nächste State, der angesprungen werden soll.

Indexierte Speichervariablen:

Vom prinzipiellen Aufbau her ist diese Version nahezu identisch mit einer normalen *functional Global Variable*. Der Unterschied liegt darin, dass intern ein Array abgespeichert wird. Von ausserhalb kann mittels eines Indexes auf den Wert zugegriffen werden. *slideSelection* wird nur bei der Initialisierung verwendet und initialisiert den internen Array auf 36 oder 60 Datenfelder.

Man kann also sagen, es entspricht in etwa einem gekapselten Array einer modernen Programmiersprache.

Einen solchen Speichertyp zu haben, bringt für die Automatikmessung sehr grosse Vereinfachungen, da die Daten an einem Ort gebündelt liegen und der Zugriff darauf kontrolliert und einfach zu handhaben ist.

init-state:

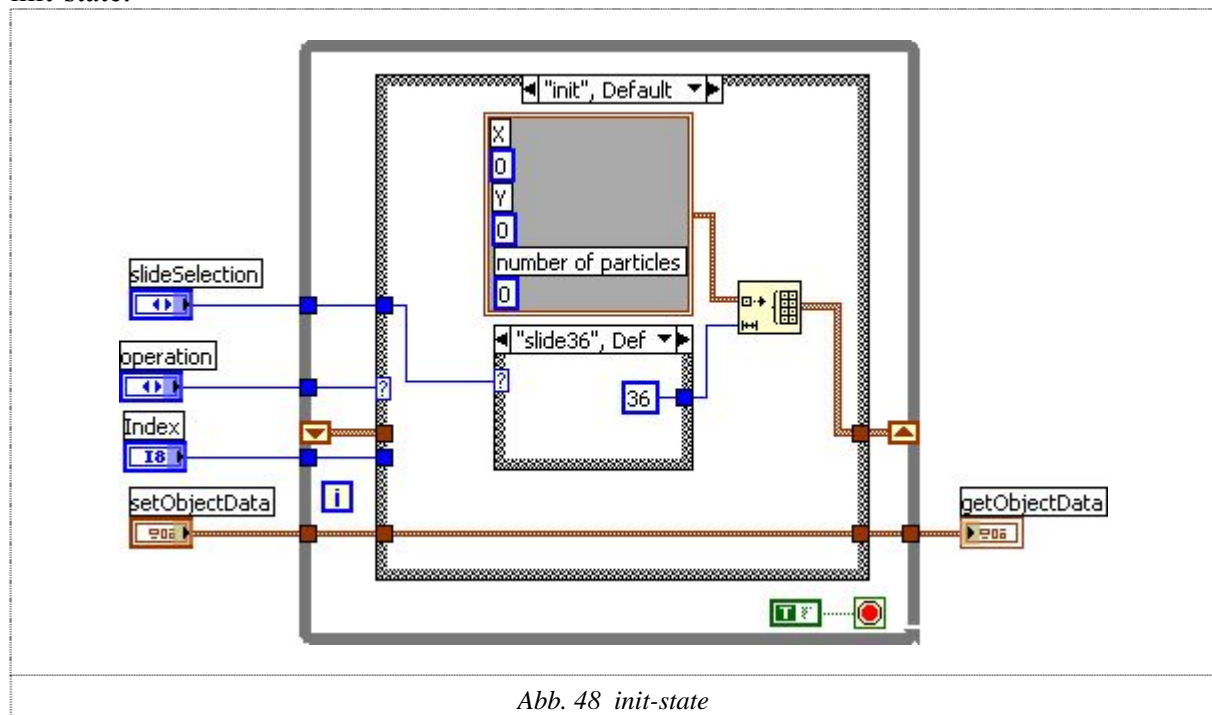


Abb. 48 init-state

Der interne Array wird auf 36 oder 60 Datenfelder initialisiert. Zugleich werden alle Datenfelder mit den Initialisierungswerten versehen.

set-state:

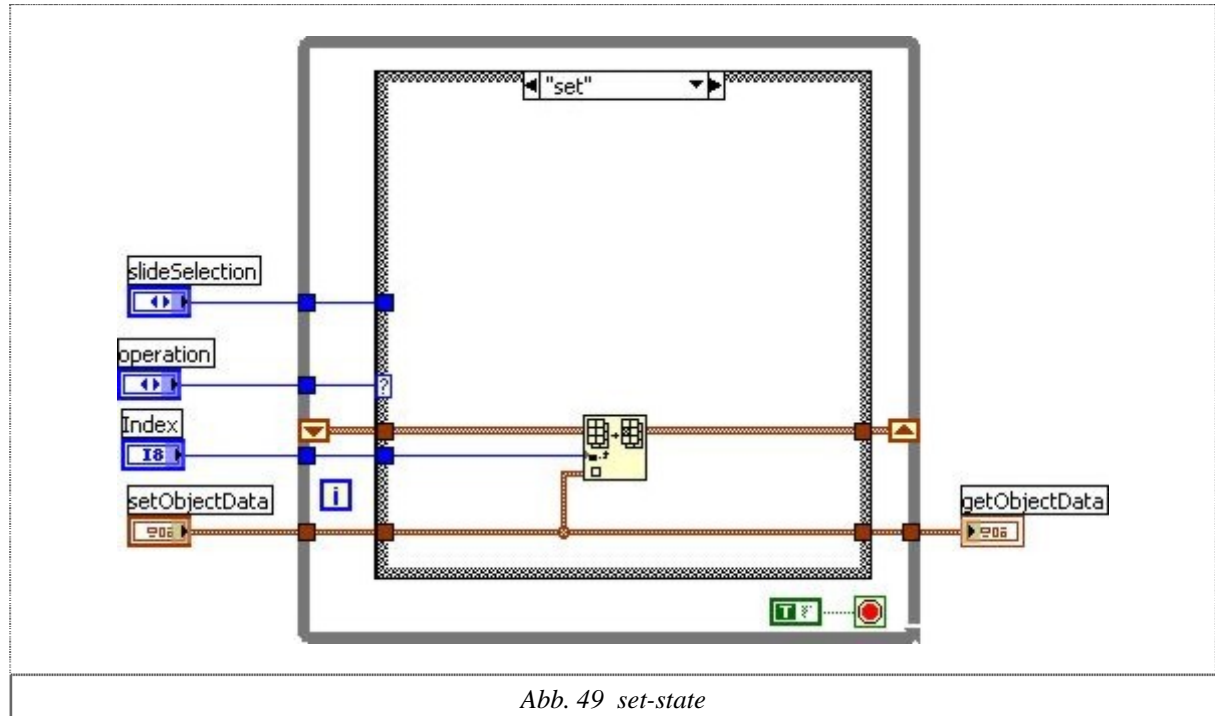


Abb. 49 set-state

Durch Übergabe des Index wird der zu speichernde Wert an die richtige Stelle im Array abgelegt.

get-state:

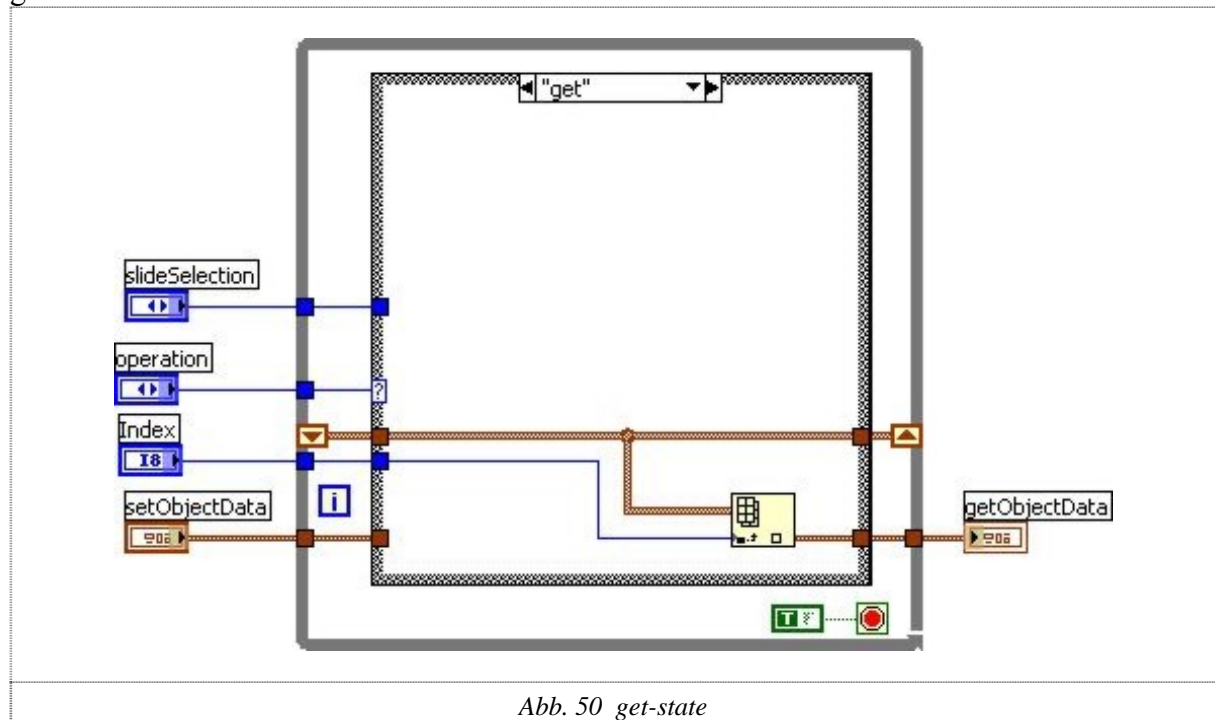


Abb. 50 get-state

Mittels Index wird auf eine Array-Speicherzelle zugegriffen und deren Wert zurückgeliefert. Intern bleibt aber der ganze Array gespeichert.

10.7 Typedefinitionen

LabView lässt das Erstellen so genannter Typedefinitionen zu. Ein Grundbaustein von LabView (z.B. ein Enum) kann bearbeitet und dann als *strict typedef* gespeichert werden. Dieser erscheint dann im Projektextplorer (in unserem Projekt im Unterordner *typedef* abgelegt) als *.ctl-Datei. Wird von nun an dieser personalisierte Baustein verwendet, so wird auf genau diese Datei zugegriffen. Somit ist es nicht möglich, einen Fall aufzurufen, der nicht vorgesehen ist, da nur die in der Typedefinition vorgesehenen Werte gewählt werden können! Werden nachträglich noch Änderungen ausgeführt, so ist nicht jeder Baustein einzeln, sondern nur die *typedef*-Datei zu editieren.

10.8 Eventstruktur

LabView verfügt über eine Eventstruktur. Hier wird interruptgesteuert auf ein Ereignis gewartet. Sobald ein Ereignis auftritt, wird der entsprechende Fall aufgerufen und ausgeführt. Es ist nötig, die ganze Struktur in eine while-Schleife zu verpacken, so dass nach dem Abfangen eines Event die Struktur neu gestartet wird!

In unserem Projekt verwenden wir als Aufrufereignisse Benutzereingaben (also das Drücken von Buttons, Eingeben von Textfeldern etc.) und Timeouts.

Reaktion auf Benutzereingabe

Über den Eventhandler wird bestimmt, worauf reagiert werden soll. Im Beispiel unten geschieht dies auf eine Wertänderung (Value Change) des Buttons (*btnAutomeasure*). Zudem stellt die Eventstruktur gewisse Werte (links in der Tabelle) zur Verfügung (im Beispiel wird keiner verwendet). Vor allem *NewVal* und *OldVal* sind sehr hilfreich.

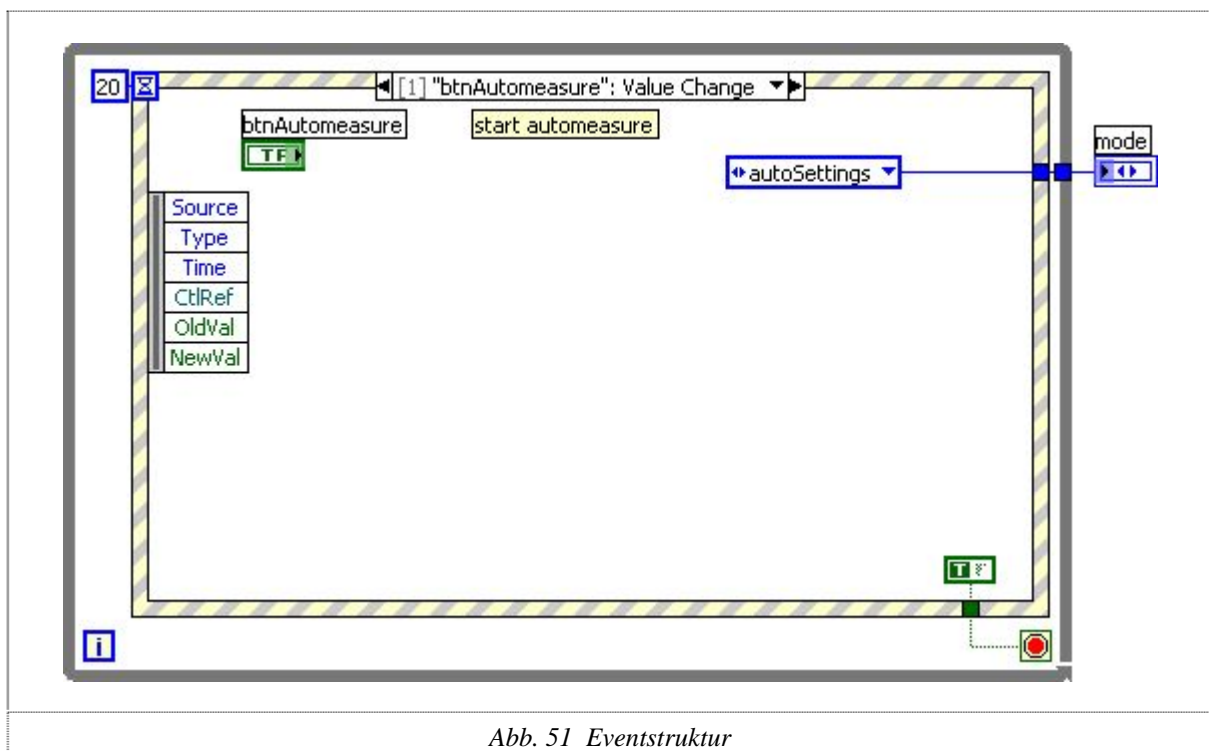


Abb. 51 Eventstruktur

Timeout

Auch hier muss über den Eventhandler das Auslöseereignis gewählt (also Timeout) werden. Der "kleinen blauen Sanduhr" am linken oberen Rand der Struktur wird übergeben, nach wie viel Zeit in Millisekunden das Timeout ausgelöst werden soll. Im Beispielfall unten geschieht dies nach 20 Millisekunden. Logischerweise wird bei jedem Timeout der Inhalt der Event-Struktur ausgeführt!

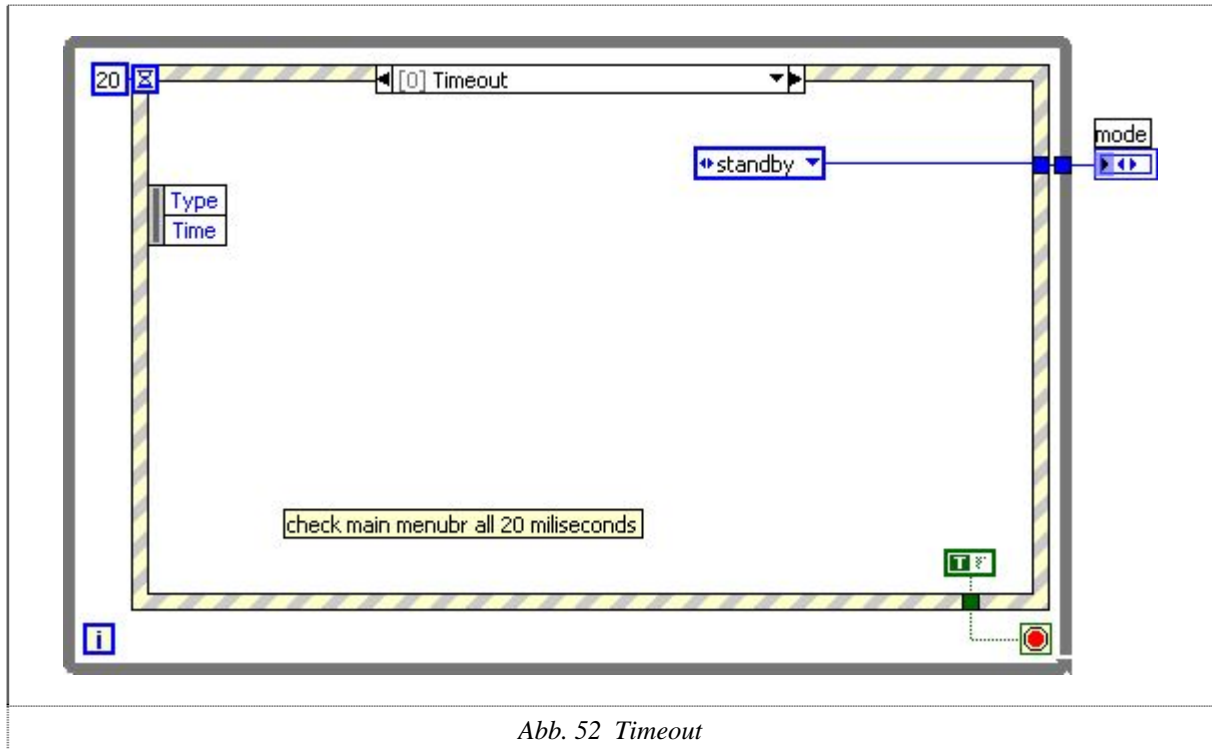


Abb. 52 Timeout

10.9 Fehlercodes

Wer sich etwas intensiver mit LabView beschäftigt, der merkt, dass LabView über eine eigene Fehlerstruktur verfügt. Ein Fehler ist grundsätzlich ein Cluster von 3 Datentypen: ein Boolean, der den Fehler anzeigt (*false* = kein Error und *true* = Error), eine Integerwert, der den effektiven Fehler bezeichnet und ein String, welcher den Fehler genauer spezifiziert.

National Instruments sieht einen Bereich vor, in welchem Benutzer selbst erzeugte Fehler anlegen können. Dieser Bereich ist von *National Instruments* vorgegeben und befindet sich zwischen den Fehlercodenummern 5000 und 9999.

Angefügt ist eine Auflistung der Fehler und ihrer "Erzeuger-VI's". Die Beschreibung ist eine Übersetzung ins Deutsche und entspricht nicht eins zu eins dem Programmstring, da dieser, wie die Programmiersprache, in Englisch geschrieben ist.

Errorcode	Beschreibung	Sender-VI
6000	Motorfokus (relatives Bewegen) ausserhalb der Limiten	<i>FocusControl.vi</i>
6001	Motorfokus (absolutes Bewegen) ausserhalb der Limiten	<i>FocusControl.vi</i>
6002	Zentrieren nicht möglich, da kein Objekt detektiert	<i>autocenter.vi</i>
6003	Rollerror: Fehler während dem Ausrichten der Roll-Achse	<i>autotilt.vi</i>
6004	Pitcherror: Fehler während dem Ausrichten der Pitch-Achse	<i>autotilt.vi</i>
6005	Fokussieren nicht möglich (Zu viele Schritte benötigt)	<i>autofocus.vi</i>

11 Erfahrungen

11.1 Umwege und Probleme

Anfangs sahen wir eine Problemquelle im Autofokus, da wir überhaupt nicht abschätzen konnten, wie weit uns *Leica* behilflich sein würde. Schliesslich konnten wir jedoch auf eine grössere Unterstützung zählen, als wir zu Beginn erhofft hatten. Mit dieser Unterstützung kam die ganze Ansteuerung des Motorfokus sehr schnell zum Funktionieren.

Beim Tisch hatten wir mehrere Probleme mit der Konstruktion zu bewältigen. Der Slidehalter war sehr ungenau und die Slides rutschten bei Kippbewegungen heraus. Auch war der ganze Halter so platziert, dass nicht der komplette Slide abgefahren werden konnte. Aus diesen Gründen konstruierten wir zusammen mit der mechanischen Werkstatt einen komplett neuen Halter. Da der Schwenktisch keine Möglichkeit für die Befestigung am Mikroskop besass, mussten wir diese zuerst evaluieren und anschliessend eines unserer Ergebnisse umsetzen, bevor der Tisch brauchbar funktionieren konnte.

Das Ausführen der Software unter der LabView-Runtime brachte erstmals richtige Probleme mit sich. Am Anfang konnte uns niemand sagen, warum die Software kein Bild lieferte und der Rest einwandfrei funktionierte. Wir versuchten das Problem auf alle möglichen Arten, wie z.B. Installieren der kompletten Treiber, zu beheben. Das brachte uns nicht weiter. Auch der Support von LabView konnte nicht weiterhelfen. Erst nach einer internen Abklärung von *National Instruments* kam heraus, dass die *Vision*-Komponenten eine kostenpflichtige *Vision*-Runtime benötigen. Leider wurde diese Runtime für LabView 8.0 nicht mehr ausgeliefert, da sie wahrscheinlich zu viele Fehler besass. So mussten wir zum Schluss unser Projekt nach LabView 8.20 konvertieren.

11.2 Erreichte Ziele

Da wir die Probleme, die sich uns stellten, lösen konnten, erfüllten wir die Aufgabenstellung komplett. Darüber hinaus wurde das Projekt mehrere Male mit Zusätzlichen erweitert, so dass am Schluss der Funktionsumfang weit aus grösser geworden ist, als die Aufgabenstellung zu Beginn lautete.

Folgende Ziele wurden erreicht:

- Ansteuerung des *Leica*-Mikroskopes über die serielle Schnittstelle
- Evaluation verschiedener Möglichkeiten zur Fokussierung
- Entwicklung eines Algorithmus, der sauber und zuverlässig Fokussieren kann.
- Ansteuerung des Schwenktisches
- Erstellung eines Algorithmus, der die Mikrofossilien, egal in welcher Lage (Kiel, Spiral, Umbilikal), ausrichtet
- Erstellen eines Bedienerprogramms, welches ein Einzelmessmodus und eine Automatikmessung erlaubt. In der Einzelmessung wird dem Benutzer grosse Freiheiten, aber auch die notwendige Unterstützung geliefert. Die Automatikmessung bearbeitet ganze Messstreifen, ohne dass der Anwender noch Eingriffe vornehmen muss.

11.3 Feedback

Als wir die Diplomarbeit angenommen hatten, freuten wir uns auf eine interessante Arbeit. Wir konnten uns aber nur schwer vorstellen, wie weit und tief die Aufgabe gehen würde, da sie Themen behandelte, welche nicht direkt zu unseren Fachgebieten zählten. Dies änderte sich jedoch, als wir effektiv zu planen begannen und mit Herrn Dr. Michael Knappertsbusch den gegenseitigen Erfahrungshorizont austauschten. Es war für uns sehr interessant und lehrreich, mit ihm zusammenzuarbeiten, da er aus einer anderen Fachrichtung kommt und somit das Projekt von einem anderen Blickwinkel her betrachtete. Wir konnten gegenseitig unsere Erfahrungen in Bezug auf die Bildverarbeitung vergleichen. Sehr begeistert waren wir darüber, wie aktiv er sich an der Entwicklung beteiligte. Immer wieder brachte er Ideen, Optionen und Punkte ein, die uns zum Weiterdenken anregten.

Die Zusammenarbeit in unserem Partnerteam klappte hervorragend, so dass wir unsere Arbeiten optimal aufteilen konnten. Dies ermöglichte ein zügiges und zielstrebiges Arbeiten.

Wir freuen uns, dem Naturhistorischen Museum einen ausgereiften und funktionsfähigen Aufbau zum Untersuchen von Mikrofossilien übergeben zu können. Mit unserer Diplomarbeit kann Herr Dr. Michael Knappertsbusch viel Zeit bei der Untersuchung der Fossilien sparen und somit seine Forschungen vorantreiben.

11.4 Ausblick

Motorisierte Vergrößerungswahl

Für den vollautomatisierten Ablauf wäre es sinnvoll, wenn die Vergrößerung vom PC aus angesteuert werden könnte. So müsste man nach dem ersten Durchlaufen des Slides die Vergrößerung nicht mehr manuell verstellen, und es könnten kleinere und grössere Fossilien auf einem Slide vermessen werden. Dies würde ein Schrittmotor mit der entsprechenden Ansteuerung und dem PC-Interface benötigen.

Rotation in der Spiralansicht

Die Drehung und Ausrichtung der Schale in der Spiral- oder Umbilikal Position ist noch ungelöst. Die bisherige manuelle Methode (horizontale Verbindung Einschnitt-Apex auf $Y=240$) ist bildanalytisch sehr schwierig. Ein Pattern Matching wird als Möglichkeit vorgeschlagen. Dazu könnte man z.B. mehrere, präzis vorpositionierte Exemplare als Vergleichsobjekte verwenden und das Zielobjekt korrelieren lassen bis die beste Übereinstimmung gefunden wurde. Der Ansatz über mehrere Exemplare (anstatt nur eines zu nehmen) gewährt mehr Information bei der vorhandenen Formvariabilität der Schalen. Dieser Ansatz wird im Moment nicht implementiert, weil er zum jetzigen Zeitpunkt zu aufwändig ist. Das heisst in der Spiral- und Umbilikalansicht wird derzeit keine rotatorische Korrektur vorgenommen.

Polarisierte Ringbeleuchtung

Zur Beleuchtung: Bei den randständigen Exemplaren im Slide (ganz links oder ganz rechts) kann es bei der Kipp-Bewegung Probleme geben, wenn Seitenbeleuchtung (Schwanenhals) eingesetzt wird. Dann nämlich wird jeweils nur einseitig beleuchtet, weil die Slidewanne das Objekt in den Lichtschatten stellt. Somit funktioniert der Autofokus nicht mehr optimal. Ideal wäre, wenn auch die Ringleuchte mit einem Polarisationsfilter versehen werden könnte. Eine Lösung wäre ein passendes Metallrohr als Übergangsstück zur Lichtquelle. Das Rohr weist einen Schlitz auf, in den man einen Pol-Filter (Folie) drehbar einlagern könnte. Somit wären die Vorteile beider Beleuchtungen kombiniert.

Bildverarbeitung

Die jetzige Bildauswertung wird mit separater Software durchgeführt. Es wäre jedoch möglich, die komplette Auswertung direkt im Programm mittels *Vision*-Komponenten zu realisieren. Dies könnte direkt nach der Bilderfassung oder auch am Ende der Messung mit den abgespeicherten Bildern geschehen. Zum Vornherein müsste man jedoch zuerst abklären, ob *Vision* mächtig genug ist, um die bisherigen Programme zu ersetzen.

12 Danksagung

Unser besonderer Dank gilt:

- Allen Lesern für das Interesse an unserer Arbeit
- Herrn Prof. Dr. Jean Eisenecker für die Projektbetreuung
- Frau Martine Lefebvre für das Amt als Expertin
- Herrn Christoph Biel für die Unterstützung im Labor
- Herrn Dr. Michael Knappertsbusch für den Auftrag
- Herrn Michael Bogucki für die Unterstützung in LabView
- Herrn Ronny Bachmann fürs das Einrichten der PC's
- Herrn George Hasler für die mechanischen Arbeiten
- Herrn Daniel Binggeli für die Erstellung des Schwenktisches

13 Erklärung

Hiermit erklären wir, dass wir die gesamte Arbeit selbstständig und nur mit den im Quellenverzeichnis angegebenen Hilfsmitteln geschrieben haben.

Andreas Herzig

Lukas Schmutz

Muttenz, 10. Januar 2007

14 Quellenverzeichnis

14.1 Linkliste

LabView

<http://www.ni.com/labview>

National Instruments

<http://www.ni.com>

Leica

<http://www.leica-microsystems.com/>

Wikipedia

<http://www.wikipedia.ch/>

Naturhistorisches Museum Basel

<http://www.nmb.bs.ch>

Technologie der 3D-Darstellung

http://www.digitalmikroskop.de/microscope_details/history.html

14.2 Semesterarbeiten

Automatisierung der Mikrofossilorientierung 05, A. Erni und C. Degen

14.3 Literatur

A Software Engineering Approach to LabView, Prentice Hall

LabView Basics I: Introduction, National Instruments

LabView für Studenten, Pearson Studium

15 Anhang

- Kurzbeschreibung LabView
- Installationsanleitung
- Bedienungsanleitung
- Laborjournal