

Negativsichtstation

Projektdokumentation

7. Januar - 8. Februar 2013

<https://github.com/JohannesBiermann/NegativeViewer>

Johannes Biermann
code@johannesb.eu

Staatliche Akademie der Bildenden Künste
Am Weißenhof 1
70191 Stuttgart

Abstract

The aim of this project is to construct an easy to use station for viewing photographic negatives and slides.

The station requires a digital camera but should be independent of a certain kind of camera. Originally it had been planned to be used with the common HDMI connection. But because of defective hardware it had to be changed to USB.

A computer program has been developed which works with webcams as well as with digital single lens reflex cameras. This program enables the user to invert the picture and to apply different functions like automatic cropping and rotating. For example the image can be saved for a photo database or high resolution digital copy.

Inhaltsverzeichnis

1. Einleitung	1
2. Ursprüngliches Konzept	1
2.1. Neues Konzept.....	1
3. HDMI Interface	2
4. Versuchsaufbau	2
4.1. Repräsentativ	2
4.1.1. Umbau des Stativs auf eine Standard-Kamerahalterung	4
4.1.2. Adaptieren des Leica M Anschluss	4
4.1.3. Beleuchtung.....	4
4.1.4. Versuchsaufbau.....	5
5. Software	6
5.1. Programmiersprache.....	6
5.2. Frameworks	6
5.2.1. OpenCV	6
5.2.2. GPhoto2	7
5.2.3. Qt.....	7
5.3. Linux als Betriebssystem	7
5.4. Bildverarbeitungsalgorithmen.....	8
5.4.1. Invertieren des Bildes	8
5.4.2. Konvertieren in Graustufen.....	8
5.4.3. Tonwertanpassung	9
5.4.4. Schärfen des Bildes.....	10
5.4.5. Automatisches Freistellen	11
5.5. Drehen des Bildes.....	12
5.6. Oberfläche des Programms	13
5.7. Lizenzform des Programms	15
5.8. Inbetriebnahme des Programms.....	15
5.9. Installation auf ein eigenes Betriebssystem.....	17
6. Auswertung und Test der Kamerasysteme	17
6.1. Webcam Logitech C920.....	17
6.2. Camcorder Canon HV30.....	17
6.3. Spiegelreflexkamera Nikon D300.....	18
7. Fazit und Ausblick	19
Anhang A: Stichwortartige Beschreibung der Kameraeinstellungen	a
Anhang B: Fragen und Antworten zu Problemen mit der Kamera und dem Programm	b
Anhang C: Bibliotheken und Entwicklungsumgebung installieren	c
Anhang D: Installation auf ein eigenes Betriebssystem	f
Anhang E: Abbildungsverzeichnis	g
Anhang F: Literaturverzeichnis	g
Anhang G: Projektplan	Fehler! Textmarke nicht definiert.

1. Einleitung

Viele Fotografien liegen als Negative oder Dias vor. Um sie zu sichten, legt man sie auf ein Leuchtpult. Die Kleinbild-Dias haben eine relativ kleine Bildfläche, die das genaue Betrachten erschwert. Deshalb müsste man sie für eine genauere Inhaltserschließung scannen. Die älteren Schwarzweiß-Glasplattenegative haben häufig ein größeres Format. Hier erschwert jedoch das invertierte Bild die Erschließung. Auch dieses müsste man scannen und als Positiv umwandeln oder einen neuen Abzug anfertigen.

Aus diesem Grund entstand die Idee, ein vielseitig einsetzbares Sichtgerät für durchscheinende Fotografien und Negative zu konstruieren. Das Gerät soll möglichst flexibel in Bezug auf die verwendete Kamera sein. Für eine möglichst kostengünstige Realisierung der Sichtung sollen verfügbare Standardteile eingesetzt werden.

Mit der Software soll der Benutzer die Negative leicht invertieren und als Einzelbild exportieren können. Das Standbild einer Videokamera oder Webcam ersetzt zwar keinen guten Scan oder Abfotografie, aber für die Metadatenererschließung und Bestandssichtung ist es ausreichend.

Durch die optionale Verwendung einer Spiegelreflexkamera lässt sich das Sichtgerät auch zum Anfertigen von hochwertigen Digitalisierungen verwenden.

2. Ursprüngliches Konzept

Das Konzept des Sichtgeräts besteht darauf, ein Durchleuchtpult mit einem Stativ als mechanische Basis für die Sichtung zu nutzen. Das System ist vom Kamertyp variabel. Es kann mit einer günstigen Webcam, einer Videokamera oder einer digitalen Spiegelreflexkamera betrieben werden.

Um das Bildsignal zu erfassen, wird der HDMI Anschluss verwendet. Dieser Anschluss ist mittlerweile Standard in allen Kameras. Die Webcam wiederum läuft über USB. Die Spiegelreflexkamera wird per USB und HDMI angeschlossen. Der HDMI Anschluss dient zur Erfassung des Live-Bilds und über USB wird die Kamera vom Computer aus gesteuert.

2.1. Neues Konzept

Aufgrund defekter HDMI Interfaces (siehe Abschnitt HDMI Interfaces) musste dieser Teil des Konzepts aufgegeben werden. Daher wird eine USB-Webcam als günstige Kamera und eine (semi-) professionelle Spiegelreflexkamera mit Liveview verwendet, die das Bild per USB an den Rechner weitergeben kann.

3. HDMI Interface

Zum Anschluss an den Computer wird ein HDMI Interface benötigt, welches das HDMI Signal in ein computerlesbares Format umwandelt. Damit dies klappt, reicht das Interface den Videostrom an die Standard-Multimediaschnittstelle des Betriebssystems weiter.

Momentan existieren nicht viele HDMI Interfaces am Markt. Und die meisten davon sind nur mit einem Videoschnittprogramm wie Premiere oder Finalcut nutzbar. Sie umgehen die Multimediaschnittstelle des Betriebssystems. Daher sind solche Interfaces für ein eigenes Programm unbrauchbar.

Noch rarer sind Interfaces, die per USB 3.0 arbeiten. Die Modelle von Blackmagic eignen sich nur für spezielle Mainboards für Desktoprechner, somit stellt sich die Grundsatzfrage, warum dann überhaupt ein externes Interface verwendet wird.

Momentan bietet als einziger Hersteller Epiphan universelle USB 2 und 3 Lösungen für den Computer an, die sich für eine eigene Softwareentwicklung nutzen lassen. Daher wurde das Interface Epiphan DVI2USB angeschafft. Diese Geräte sind für OEM und Industrieanwendungen vorgesehen und mit ca. 700 – 1000 Euro nicht sonderlich preiswert. Allerdings lassen sich die Geräte unter allen gängigen Betriebssystemen wie Windows, Mac und Linux betreiben.

Leider ist das für diesen Zweck neu gekaufte Interface DVI2USB bereits bei der Ankunft defekt und kann nicht so schnell umgetauscht werden, da es von Kanada geliefert wurde.

Die zweite Option ist die interne PCI Express Steckkarte von Blackmagic Design mit den Modellnamen Intensity Pro. Diese Karte unterstützt auch die Standardmultimediaschnittstellen. Die Karte ist bereits im Studiengang vorhanden. Leider ist auch diese Karte defekt.

Aufgrund der beiden defekten HDMI Interfaces musste der ursprüngliche Plan, HDMI als Standardschnittstelle zu verwenden, aufgegeben werden. Daher konzentriert sich das Projekt auf die Webcam und die Steuerung der Spiegelreflexkamera per USB.

4. Versuchsaufbau

Dieses Kapitel beschreibt den Aufbau der Sichtstation und die verwendeten Materialien. Als Basis dienen ein älteres Reprorativ und ein Leuchtpult.

4.1. Reprorativ

In der Akademie ist bereits das Reprorativ Leitz Reprovit IIa aus den sechziger oder siebziger Jahren vorhanden. Am Arm ist eine Kamerahalterung vorhanden, die sich demontieren lässt. In dieser Halterung befindet sich ein Objektiv, dessen Fokus sich per Balgen verstellen

lässt. Zur Scharfstellung dient ein Lichtschacht. Nach der Scharfstellung wird die Vorrichtung vom Sucher zur Kamera verschoben, damit die Aufnahme angefertigt werden kann. Als Kameraanschluss verfügt die Objektivhalterung über ein Leica M Bajonett.

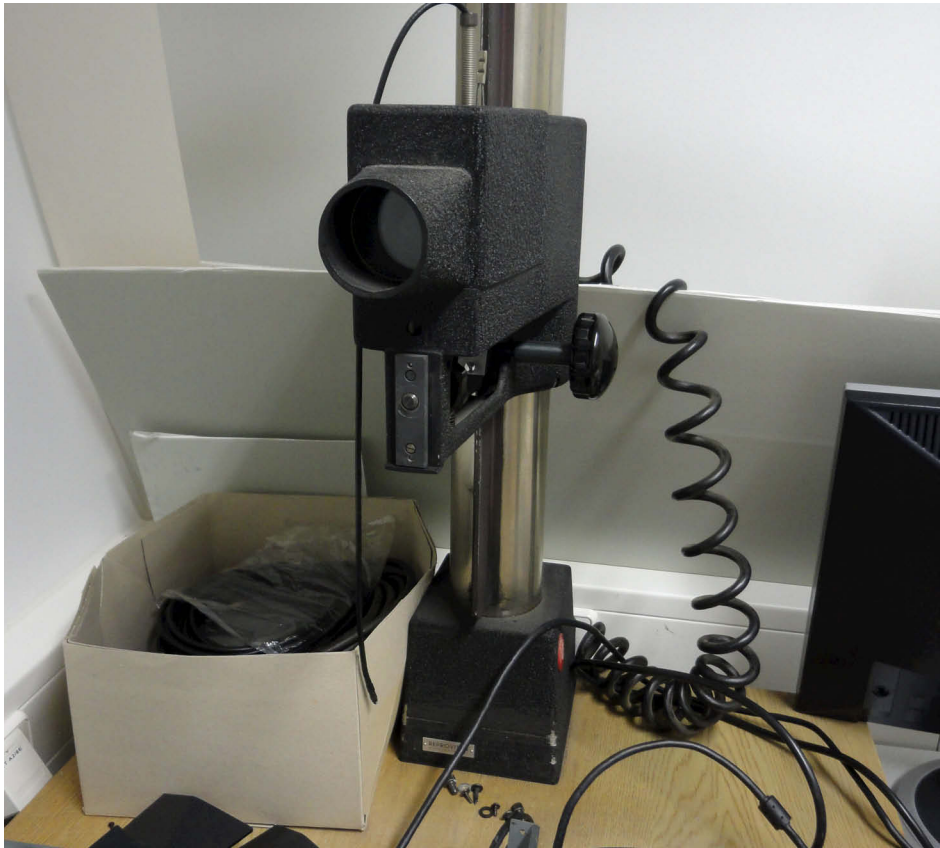


Abbildung 1: Reprostativ mit demontierter Kameraaufnahme

Bei der Abbildung ist bereits die Kamerahalterung entfernt worden. Jeweils seitlich wird die Beleuchtung für die Reprofotografie befestigt. Diese ist vorhanden, aber demontiert. Sie wird für das Sichtgerät nicht benötigt.



Abbildung 2: Halterung für die Reprokamera (Ansicht frontal und von oben)

4.1.1. Umbau des Stativs auf eine Standard-Kamerahalterung

Die Kamerahalterung einschließlich des Objektivs lässt sich am Reprostativkopf mit einer Rändelschraube leicht entfernen. Die Schraube ist in Abbildung 1 zu sehen. Nach Abmessen der Schraube hat sich ergeben, dass es sich um ein 3/8 Zoll UNC Gewinde handelt.

Daher kann ein Standard-Stativkopf mit 3/8 Zoll Gewinde montiert werden. Der Stativkopf selbst besitzt ein 1/4 Zoll Gewinde, wie es bei den meisten Kameras und Videokameras zum Einsatz kommt. Es lassen sich somit leicht eine Vielzahl von Kameras montieren.

4.1.2. Adaptieren des Leica M Anschluss

Als Alternative zum Anbringen eines Stativkopfs anstatt des Repr-Kamerasystems könnte auch ein Leica M Adapter für eine spiegellose Systemkamera verwendet werden. Somit wäre die vollständige Funktionalität des Reprostativs inklusive Lichtschachtsucher nutzbar.

Spiegellose Systemkameras sind momentan im Trend und werden von den meisten Kameraherstellern angeboten. Aufgrund des geringeren Aufmaßes der Objektive im Vergleich zu Spiegelreflexkameras ist es leicht möglich, Objektive mit Leica M Anschluss zu adaptieren. Die Reprereinheit besitzt ein Leitz Wetzlar Focotar 1:4.5/50 Objektiv.

Diese Möglichkeit wird in diesem Projekt nicht genutzt. Es wäre aber sicherlich interessant, eine Olympus Pen oder eine Panasonic Lumix Micro-Four-Thirds Kamera mit einem Adapter zu montieren und die Qualität des Objektivs zu beurteilen. Aufgrund des Cropfaktors von 2 des Micro-Four-Thirds Standards ergibt sich eine tatsächliche Brennweite wie ein 100mm Objektiv beim 24x36mm Kleinbildformat.

4.1.3. Beleuchtung

Als Durchlichtquelle wird ein kleines Leuchtpult verwendet, das bereits vorhanden ist. Damit die helle Fläche nicht die Belichtungsmessung der Kameras stört, wurden schwarze Passepartouts für die unterschiedlichen Größen der Negativmaterialien angefertigt. Bei Versuchen hat sich dann aber herausgestellt, dass die Passepartouts keinen so großen Effekt auf die Abbildungsqualität haben. Daher kann meist auch ohne Passepartouts gearbeitet werden, was der Anforderung eine leichte und schnelle Sichtstation zu schaffen, entgegenkommt.

Bei den ersten Tests hat sich gezeigt, dass das Flackern der Leuchtstoffröhren die Kameras stört und Interferenzmuster hervorrufen. Das 100 Hertz Flackern entsteht durch die Netzfrequenz. 100 Hertz daher, dass Leuchtstoffröhre auch bei den Halbperioden der Wechselspannung Licht erzeugen ($2 \times 50 \text{ Hz} = 100 \text{ Hz}$).

Leuchtstoffröhren benötigen ein Vorschaltgerät, sonst leuchten sie nicht. Dazu wird zum Start eine hohe Zündspannung sowie ein Bauteil, das den Strom begrenzt, benötigt. Das Flackern lässt sich dadurch beseitigen, dass ein elektronisches Vorschaltgerät verwendet wird. Elektronische Vorschaltgeräte arbeiten mit einer Betriebsfrequenz von mehreren Kilo-

hertz. Somit lässt sich durch Austausch des Vorschaltgeräts das Flimmern vermeiden. Das Gerät war zum Projektzeitpunkt nicht lieferbar, daher wurde die Beleuchtung provisorisch mit Leuchtstoffröhren gelöst, die per Batterie betrieben werden.

Anstatt des Leuchtpults können auch andere homogene Flächenleuchtquellen verwendet werden. Insbesondere ist ein LED Panel mit Diffusor denkbar oder eine LED Fläche, die seitlich leuchtet und das Licht per Spiegel verteilt wird.

4.1.4. Versuchsaufbau

Diese Aufnahme zeigt den fertigen Versuchsaufbau mit dem Reprostativ, dem Stativkopf und dem Leuchtpult. Dieser Aufbau dient als Basis für alle folgenden Versuche.

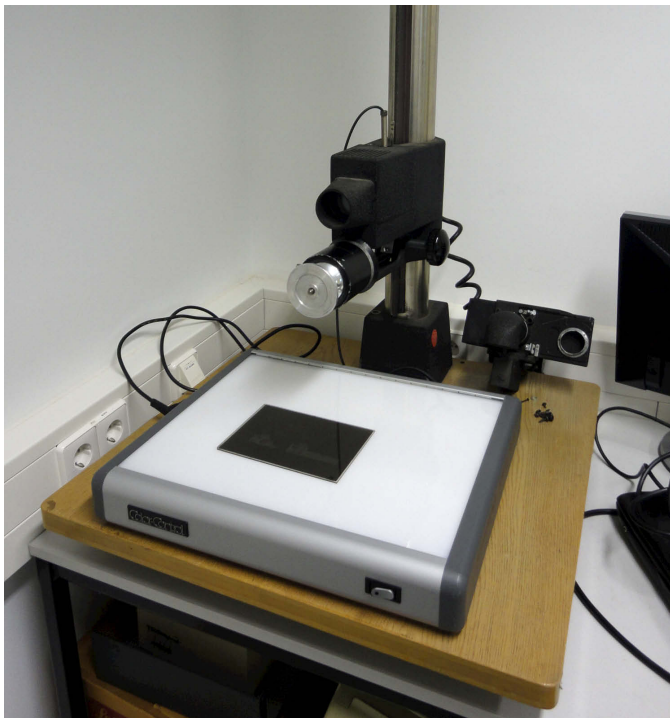


Abbildung 3: Erster Versuchsaufbau mit Stativkopf und Leuchtpult

5. Software

Damit man vielfältige Möglichkeiten hat, Negative und Dias zu sichten, wurde eine Software entwickelt, die verschiedene Bildmanipulationen auf das Sichtbild ausführen kann.

5.1. Programmiersprache

Das Negativsichtprogramm ist in C++ programmiert. C++ bietet eine sehr hohe Geschwindigkeit und es können sehr gut Programme mit grafischer Oberfläche programmiert werden. Außerdem erlaubt C++ den direkten Zugriff auf die Hardware, was bei anderen Hochsprachen nicht der Fall ist. Die hohe Geschwindigkeit wird aufgrund der hohen Datenrate von 70-190 MB/s bei Full HD Auflösung benötigt.

5.2. Frameworks

Damit die Anwendung in vorgegebener Zeit entwickelt werden konnte, wurde auf verschiedene Frameworks zurückgegriffen, die verschiedene Funktionen haben. Dies sind insbesondere OpenCV, GPhoto2 und Qt.

5.2.1. OpenCV

Als Bildverarbeitungsframework wurde Open Source Computer Vision Library – kurz OpenCV – ausgewählt (<http://opencv.org>). Das Framework wurde von Intel entwickelt und ist 1999 entstanden. Die Lizenzform des Frameworks ist BSD, somit kann es für kommerzielle und nicht kommerzielle Anwendungen eingesetzt werden.

Es ist auf die Verarbeitung von Echtzeitdaten spezialisiert, kann aber auch mit Standbildern arbeiten. Das Framework bietet bereits viele Algorithmen zur Bildverarbeitung und Manipulation. OpenCV selbst ist in C++ geschrieben und bietet Schnittstellen zu anderen Programmiersprachen wie C, Python und Java.

OpenCV liefert jedes einzelne Frame als dreidimensionale Matrix, die man zur Bildverarbeitung manipulieren kann. Die Matrix besteht aus: B für den Blaukanal, G für den Grünkanal und R für den Rotkanal (RGB Farbsystem). Jede Position in der Matrix enthält den Farbwert des Pixels. Die Abbildung 4 visualisiert die BGR Matrix¹.

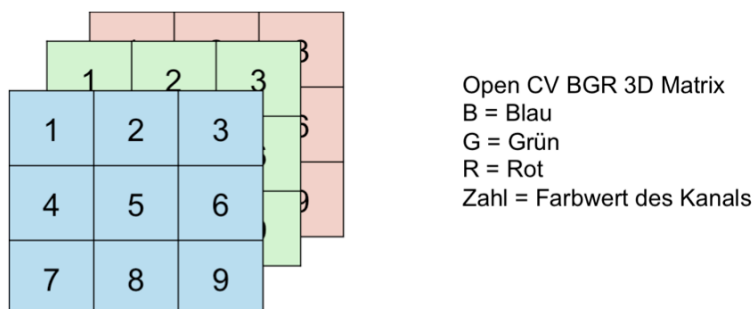


Abbildung 4: OpenCV Bildmatrix

¹ OpenCV Dev Team: Mat - The Basic Image Container, http://docs.opencv.org/doc/tutorials/core/mat_the_basic_image_container/mat_the_basic_image_container.html#matthebasicimagecontainer, abgerufen 16.02.2013.

5.2.2. GPhoto2

GPhoto2 ist eine Framework und Programm zur Ansteuerung von Digitalkameras. Die Hauptfunktion ist das Zugreifen auf die Speicherkarte in der Kamera, daher ist es Basis von vielen Bilddatenbanken unter Linux.

Darüber hinaus kann GPhoto2 auch Kameras steuern, sofern die Kamera dies unterstützt. Der große Vorteil daran ist, dass dies herstellerübergreifend funktioniert. In der Praxis hat sich gezeigt, dass die Kameras unterschiedlich gut funktionieren. Dies muss mit der zu steuernden Kamera ausprobiert werden. Leider gibt es keinen gemeinsamen Standard zur Kamerasteuerung. Jeder Hersteller besitzt seine eigenen Funktionen und benutzt zum Teil proprietäre Protokolle. Prinzipiell lassen sich hauptsächlich Kameras von Canon und Nikon steuern, die anderen Hersteller haben diese Funktionalität meist überhaupt nicht in die Kamera eingebaut.

Die Kombination von OpenCV und GPhoto2 war eine besondere Herausforderung in diesem Projekt. Und so eine Kombination ist bis jetzt wahrscheinlich auch noch nicht ausprobiert worden. Deshalb bin ich besonders stolz, dass mir dies gelungen ist.

5.2.3. Qt

Die Anwendung soll möglichst benutzerfreundlich sein. Daher besitzt die Sichtenanwendung eine grafische Oberfläche. Als Toolkit für die grafische Oberfläche wurde Qt ausgewählt (<http://qt-project.org/>). Qt ist ebenfalls in C++ programmiert. Die Besonderheit von Qt ist, dass die Anwendungen plattformunabhängig sind. Es entstehen dabei native Anwendungen. Auf der jeweiligen Plattform muss das Programm aber neu kompiliert werden. Es gibt keinen Bytecode Zwischencompiler, wie das bei Java der Fall ist.

5.3. Linux als Betriebssystem

Die Negativsichtstation-Anwendung wurde für das freie Betriebssystem Linux entwickelt. Diese Entscheidung wurde aufgrund der Spiegelreflexsteuerung und Fokussteuerung der Webcam getroffen.

Wie bereits im GPhoto2 Abschnitt erwähnt, hat jeder Hersteller seinen eigenen Standard zur Kamerasteuerung. Bei Canon und Nikon gibt es für die Kameramodelle unterschiedliche Entwicklungskits für Windows. Dies bedeutet aber, dass die Unterstützung für jedes Kameramodell entwickelt werden müsste. GPhoto2 erleichtert den Umgang durch eine gemeinsame Schnittstelle für alle Kameramodelle und Hersteller wesentlich. GPhoto2 läuft aber nur unter Unix Systemen.

Eine weitere Option wäre die Unterstützung von Mac OS X. Hier lässt sich GPhoto2 auch betreiben, Qt und OpenCV sind sowieso unter OS X lauffähig. Die Fokussteuerung der Logitech Webcam müsste aber neu entwickelt werden. Insgesamt würde eine OS X Portierung sicherlich weitere fünf Arbeitstage und viele Tests bedeuten.

Für eine Windows Version müssten etwa 50% der Funktionen ganz neu entwickelt werden und es gäbe große Einschränkungen in der Unterstützung von verschiedenen Modellen.

5.4. Bildverarbeitungsalgorithmen

Das Kapitel stellt die wichtigsten Bildverarbeitungsfunktionen und deren Algorithmen dar. Diese Funktionen zur Bildmanipulation sind alle in der entwickelten Software enthalten.

5.4.1. Invertieren des Bildes

Ein fotografisches Negativ ist ein Bild mit umgekehrten Tonwerten. Das was auf dem Negativ weiß erscheint, wird im Positiv schwarz. Somit ist der Algorithmus zum Berechnen des Positivs folgender²:

$$Positiv_{Kanal} = (L - 1) - Wert_{Negativ}$$

Diese Berechnung muss für jeden Farbkanal separat erfolgen, sofern es sich nicht um ein Graustufenbild handelt. L ist die Farbtiefe, die in der Regel 8 Bit, also 256 beträgt.

5.4.2. Konvertieren in Graustufen

Damit ein Farbbild in Graustufen umgewandelt werden kann, muss die Helligkeit des Pixels in dem jeweiligen Farbkanal festgestellt werden. Danach wird die festgestellte Helligkeit zu einem Graustufenwert umgewandelt.

Es gibt dabei verschiedene Möglichkeiten zur Umrechnung. Die Kanäle können unterschiedlich berücksichtigt werden. Einzelne Kanäle können sogar weggelassen oder gleichwertig behandelt werden.

OpenCV benutzt folgenden Algorithmus³:

1. Umwandeln des Bildes in den Y'UV Farbraum
2. Benutzung den Y Kanal als Graustufenwert

Y'UV ist ein Farbraum, der bei der Umstellung von Schwarz-Weiß zum Farbfernsehen entstanden ist. Dabei sollte sichergestellt werden, dass die Schwarz-Weiß Fernseher weiterhin auch mit dem Farbsignal funktionieren. Dazu wird das Farbsignal in den Y'UV Farbraum gewandelt. Y'UV trennt die Farbigekeit von der Helligkeit. Y ist die Luminanz und U+V sind die Chrominanz Signale. Das SW Gerät zeigt nur das Y Signals, welches die Helligkeit (= Graustufen) bedeutet. Das Farbgerät benutzt auch die UV Kanäle zum Anzeigen der Farben⁴.

Die Umrechnungsformel lautet:

$$Y = 0,299 * R + 0,587 * G + 0,114 * B$$

² Loch, G. Francis: Image Processing Algorithms Part 7: Colour Inversion And Solarisation, <http://www.dfstudios.co.uk/articles/image-processing-algorithms-part-7-colour-inversion-and-solarisation>, abgerufen 25.01.2013.

³ OpenCV Dev Team: Miscellaneous Image Transformations, http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor, abgerufen 25.01.2013.

⁴ Wikipedia, The Free Encyclopedia: YUV, <http://en.wikipedia.org/w/index.php?title=Special:Cite&page=YUV&id=533236206>, abgerufen 25.01.2013.

5.4.3. Tonwertanpassung

Ein Histogramm zeigt die Häufigkeitsverteilung von Werten an, im Fall von einem Tonwert-histogramm die Verteilung der Tonwerte. Dabei hat jeder Kanal eines Bildes ein eigenes Histogramm, welches manchmal in üblichen Bildbearbeitungsprogrammen auch zu einem Gesamthistogramm zusammengefasst wird.

Eine automatische Tonwertanpassung zieht dicht zusammenliegende Bereiche auseinander und nutzt das ganze Tonwertspektrum (0 - 255 bei 8 Bit) aus.

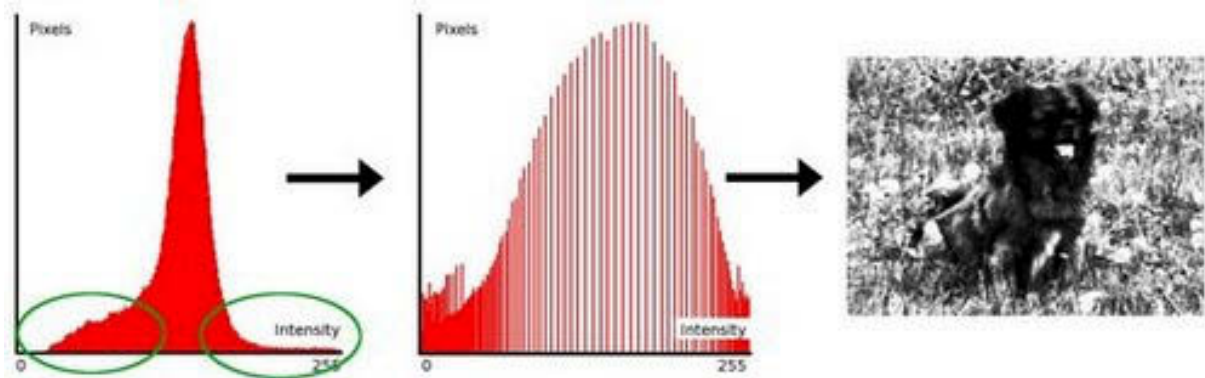


Abbildung 5: Automatische Tonwertkorrektur (Quelle: http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html)

Das Histogramm baut sich anhand der Häufigkeitsverteilung auf:

$$p_n = \frac{\text{Anzahl der Pixel mit Intensität } n}{\text{Gesamtanzahl der Pixel}} \quad n = 0, 1, \dots, L - 1$$

wobei L die Tiefe des (Farb-)Kanals ist. In der Regel ist das 8 Bit, also 256.

Dies Histogramm-Anpassung berechnet sich mit:

$$g_{i,j} = \text{floor} \left((L - 1) \sum_{n=0}^{f_{i,j}} p_n \right)$$

L ist wieder die Farbtiefe und floor rundet eine Fließkommazahl auf den nächsten Integer Wert ab (vgl. der Formeln ⁵). Die Formeln bedeuten letztendlich, dass Farbwerte mit hohem Vorkommen einen hohen prozentualen Anteil besitzen. Je höher der Prozentwert ist, desto mehr werden die Tonwerte auseinander gezogen. Farbwerte mit geringem Prozentwert werden stärker gestaucht.

Bei einem Farbbild muss die Anpassung für jeden Kanal erfolgen. Dies kann jedoch heftige Farbverschiebungen verursachen, da die sichtbare Farbe eine Kombination aus allen drei Kanälen ist. Daher wird der Farbraum des Bilds in einen Farbraum umgewandelt, der Helligkeit und Farben trennt. Nur auf den Kanal mit den Helligkeitsinformationen wird eine automatische Tonwertanpassung. Bei dem Programm wird daher das Farbbild zu dem HSV Far-

⁵ N.N.: Histogram Equalization: http://math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf, abgerufen 25.01.2013.

braum umgewandelt und nach der Tonwertkorrektur des Helligkeitskanals V wieder zum RGB Farbraum zurück konvertiert.

5.4.4. Schärfen des Bildes

Die Idee Bilder zu schärfen, basiert immer darauf, dass das menschliche Auge ein Bild als schärfer empfindet, wenn die Kontraste an den Motivkanten stärker sind. Die Algorithmen heben die Kanten der Objekte im Bild stärker hervor. Somit erscheint das Bild für das menschliche Auge schärfer.

Es gibt verschiedene Algorithmen, die diesen Effekt ermöglichen. Bekannt aus den Bildverarbeitungsprogrammen ist „Unschärf maskieren“ und schärfen mit Laplace Filter.

Bei dem Programm wurde „Unschärf maskieren“ verwendet. Der Algorithmus funktioniert so (vgl. ⁶):

- Es wird eine Kopie des Bilds erstellt und das Bild mit dem Gaußschen Weichzeichner weichgezeichnet (unschärf gemacht). Der Weichzeichner basiert auf der Gaußschen Normalverteilung. Kurz gesagt erzeugt der Filter konzentrische Kreise vom Mittelpunkt des jeweiligen Pixels mit dem Radius r . Der neue Pixelwert ist der Durchschnittswert der benachbarten Pixel. Dadurch ergibt sich insgesamt ein unscharfes Bild.
- Das unscharfe Bild wird vom Originalbild subtrahiert und man erhält eine Maske. Die Maske enthält in ein Bild, welches nur die Bildkanten enthält.
- Die Maske wird auf das Originalbild addiert. Somit verstärken sich die Kanten, während andere Bildteile gleich bleiben.

Der Laplace-Filter⁷ arbeitet ähnlich, allerdings wird kein unscharfes Bild benötigt. Die Laplace-Filter erzeugt vom Ausgangsbild ein neues Bild, welches nur die Bildkanten zeigt. Die Kopie wird wieder mit dem Original verrechnet und man erhält ein schärferes Bild.

Der Nachteil der Bildschärfung ist, dass es Säume geben kann und das Rauschen verstärkt wird. Tatsächlich wird das Bild nicht schärfer, sondern die Bildkanten werden hervorgehoben. Die physikalische Auflösung steigt mit dieser Methode nicht.

⁶ Spring, R. Kenneth u.a.: Unsharp Mask Filtering, <http://learn.hamamatsu.com/tutorials/flash/imageprocessing/unsharpmaskfiltering/index.html>, abgerufen 16.02.2013.

⁷ Fanning, David W.: Image Sharpening with a Laplacian Kernel, http://www.idlcoyote.com/ip_tips/sharpen.html, abgerufen 16.02.2013.

5.4.5. Automatisches Freistellen

Damit ein aufgelegtes Bild auf dem Leuchttisch automatisch zugeschnitten werden kann, muss die Randfläche entfernt werden. Dazu wird folgender Algorithmus benutzt:

1. Ausgangsbild, dreidimensionales Objekt, Schattenwurf



Abbildung 6: Automatisches Freistellen Ausgangsbild

Von dem Ausgangsbild wird eine Kopie des Bildes angelegt, auf Graustufen reduziert und invertiert. Soll ein schwarzer Rand (Passepartout) entfernt werden, darf nicht invertiert werden.



Abbildung 7: Automatisches Freistellen invertiertes Graustufenbild (weißer Rand entfernen)

Der überraschende Kniff erfolgt jetzt: Das Bild wird in ein Binärbild umgewandelt. Das heißt, dass nur noch die Werte Schwarz und Weiß in dem Bild vorkommen. Dazu gibt es einen Schwellwert. Der Schwellwert bestimmt, welcher Tonwert als Grenze zwischen Schwarz und Weiß gewertet wird. OpenCV besitzt eine Funktion, die ein Rechteck um ein Objekt aufspannen kann. Dazu müssen die weißen Punkte gefunden werden. Die ersten weißen Punkte des Bildes bilden die Kanten des Rechtecks, der schwarze Rand wird nicht beachtet:

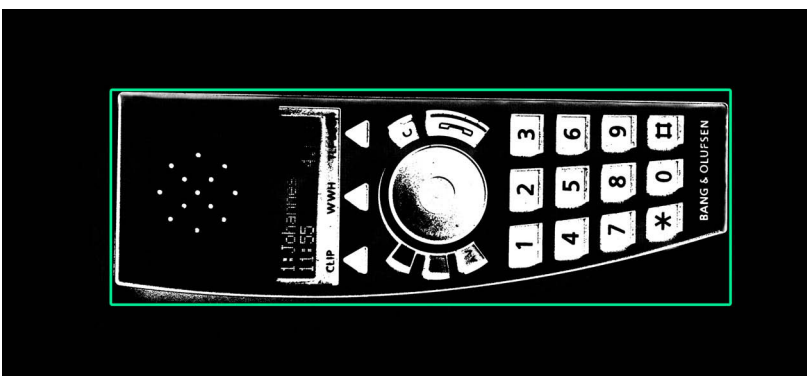


Abbildung 8: Automatisches Freistellen Binärbild mit markiertem Ausschnittsbereich

Somit erhält man die Koordinaten des aufgespannten Rechtecks im Binärbild. Bei dem Beispiel wurde ein hoher Schwellwert gewählt, damit der Schatten des Objekts möglichst nah abgeschnitten wird.

Das Originalbild wird nun anhand der ermittelten Koordinaten zugeschnitten:



Abbildung 9: Automatisches Freistellen fertiges Ergebnis

5.5. Drehen des Bildes

Das Drehen des Bildes um 180 Grad ist eine vergleichsweise einfache Operation. Um das Bild um 180 Grad zu drehen, muss die aktuelle Matrix des vorliegenden Frames gedreht werden.

Beispiel:

1	2	3	=>	9	8	7
4	5	6		6	5	4
7	8	9		3	2	1

Wobei 1, 2, 3, ... die Farbwerte des jeweiligen Pixels sind. Sofern es sich um ein Farbbild handelt, muss das Ganze für jeden Farbkanal erfolgen.

5.6. Oberfläche des Programms



Abbildung 10: Bedienoberfläche des Programms

Die Abbildung 10 zeigt die Bedienoberfläche des Programms mit dem Betrieb einer Spiegelreflexkamera. Dabei haben die Buttons folgende Funktion:

- Save: Speichert das aktuell angezeigte Bild als PNG, JPEG oder TIF ab.
- Pause: Pausiert das Bild. Der Button wird auch eingeschaltet, wenn mit „Take Photo“ ein Bild mit der Spiegelreflexkamera gemacht wurde.
- Invert/normal: Invertiert das Bild.
- Flip: Dreht das Bild um 180 Grad.
- Greyscale: Wandelt das Bild in Graustufen um.
- Sharpen: Unschärf maskieren – Bild schärfen.
- Equalize: Auto-Histogramm (Kontrastanhebung). Diese Option funktioniert mit Graustufenbild besser.
- Crop: Automatischer Zuschchnitt des angezeigten Bilds. Mit dem Regler „Crop Threshold“ kann der Toleranzbereich verändert werden. Dabei ist die linke Position weniger streng, weiter nach rechts erfolgt aggressives Freistellen.
- Crop white: Nur sichtbar, wenn der Button Crop angeklickt wurde. Crop white bedeutet, dass ein weißer Rahmen um das abgebildete Foto abgeschnitten wird. Ist „Crop white“ deaktiviert, so wird ein schwarzer Rand entfernt. Diese Betriebsart entspricht der Verwendung mit einem schwarzen Passepartout, welches das Foto ab den Rändern vollständig abdeckt.

Diese Buttons sind nur bei dem Betrieb mit einer Spiegelreflexkamera sichtbar:

Hinweis zur Verwendung: Die Spiegelreflexkamera muss wie in Anhang A beschrieben eingestellt sein, sonst funktioniert die Aufnahme nicht.

- Take Photo: Schaltet den Liveview Modus der Kamera ab und löst den Auslöser der Kamera aus. Das hochauflösende Bild wird nach einigen Sekunden in dem Fenster angezeigt. „Screen Fit“ und „Pause“ wird automatisch eingeschaltet. Screen Fit skaliert das Foto passend zur Fenstergröße.
- Autofocus: Löst den Autofokus der Kamera aus und stellt die Belichtung ein. Der Autofokus muss dabei an der Kamera eingestellt sein, sonst hat der Schalter keine Funktion. Vorsicht: Wurde ein Foto aufgenommen, kehrt die Kamera in den Liveview zurück. Daher gegebenenfalls das Foto vorher sichern.
- Screen Fit: Passt das angezeigte Foto auf die momentane Bildschirmansicht an. Wird der Button deaktiviert, erhält man eine 1 : 1 Ansicht (100%) des Fotos.

Diese Buttons sind nur bei dem Betrieb mit einer Webcam sichtbar:



Abbildung 11: Optionen bei Betrieb mit einer Webkamera

- Autofocus on: Deaktiviert oder aktiviert den Autofokus einer Webkamera. Sollte die Webkamera keinen Autofokus besitzen, so erscheint ein Hinweis. Mit dem Schieberegler ∞ --- M kann der Fokus von unendlich bis Makromodus verändert werden.
- Cam resolution: Verändert die Auflösung der Webcam. Verfügbar sind 640x480, 1280x720 und 1920x1080 Pixel. Standardmäßig wird die niedrigste Auflösung eingestellt. Sollte die Kamera die Auflösung nicht unterstützen, erfolgt ein Hinweis.
- Device number: Der Computer nummeriert die angeschlossenen Videogeräte durch. Es kann sein, dass die USB Webcam unter einer anderen Nummer geführt wird, wenn mehrere Geräte angeschlossen sind. Meist ist dies bei Notebooks mit eingebauter Webcam der Fall. Diese besitzt oft die Nummer 0. Um nun auf die externe Webcam umzuschalten, muss die Nummer auf 1 oder 2 geändert werden.

5.7. Lizenzform des Programms

Das Programm ist unter der Open-Source Lizenz GNU General Public License (GPL) Version 3 lizenziert. Dies erlaubt die Verbreitung und Änderung der Software, auch im kommerziellen Umfeld. Wird die Software weiter vertrieben oder veröffentlicht, so sind alle Änderungen offen zu legen und das eigene Programm unterliegt ebenfalls der GPL. Die Lizenz kann unter <http://www.gnu.org/licenses/gpl.html> eingesehen werden.

5.8. Inbetriebnahme des Programms

Die entwickelte Software läuft unter dem Betriebssystem Linux. Damit die Software ohne komplizierte Installation genutzt werden kann, wurde eine virtuelle Maschine erstellt, die sofort einsetzbar ist.

Als Virtualisierungsplattform kommt VMware zum Einsatz. Zum Ausführen kann VMware Workstation, VMware Player und auf dem Mac VMware Fusion verwendet werden. VMware Player ist kostenlos für Windows und Linux erhältlich. Diese Software kann von der Webseite <http://www.vmware.com/products/player/> heruntergeladen werden.

Nach der Installation von VMware Player wird die virtuelle Maschine „NegativeViewer.zip“ von der DVD auf den Computer kopiert und entpackt. Mit dem Klick auf die Datei NegativeViewer.vmx wird das virtuelle Betriebssystem von VMware gestartet.

Hinweis: Die virtuelle Maschine ist ein 64 Bit System. Daher muss der Computer ebenfalls 64 Bit fähig sein. Läuft ein 32 Bit Betriebssystem, so muss ausprobiert werden, ob es funktioniert. Dies ist abhängig vom Prozessor und PC-BIOS. Auf einem 64 Bit System funktioniert es ohnehin ohne Probleme.

Die virtuelle Maschine ist auf 2 GB Arbeitsspeicher eingestellt. Sollte dieser Wert eine zu hohe Anforderung für den Computer sein, kann das in den Einstellungen von VMware geändert werden.

Der Benutzername und das Passwort für den Benutzer lautet: viewer mit dem Passwort viewer. Das Betriebssystem startet ohne Passwortabfrage, außerdem hat „viewer“ sudo Rechte ohne Passwort.

Auf dem Desktop befinden sich folgende Verknüpfungen:

LXTerminal:

Dies ist ein Terminal zur Eingabe von Linux-Befehlen.

QT Creator:

QT Creator ist die Entwicklungsumgebung, in welcher die Software entwickelt wurde. Nach dem Anklicken öffnet sich das Projekt mit dem Quellcode. Von hier aus kann die Software weiterentwickelt und kompiliert werden.

NegativeViewer:

Startet die entwickelte Software zur Sichtung von Durchlichtfotos. Nach dem Start muss man auswählen, ob eine Webcam oder Spiegelreflexkamera als Eingabequelle verwendet werden soll.

NegativeViewerCommand:

Wie oben, es öffnet sich allerdings parallel ein Terminal. Auf dem Terminal werden Meldungen der Anwendung ausgegeben. Dies ist nützlich, sollte es Probleme mit der Kamera geben.

Verbinden der Kamera mit VMware

VMware kann direkt Geräte mit den virtuellen Betriebssystemen verbinden. Nach dem Start der virtuellen Maschine gibt es rechts oben ein Pfeil, mit der eine Menüleiste zum Einbinden der Geräte eingeblendet werden kann.

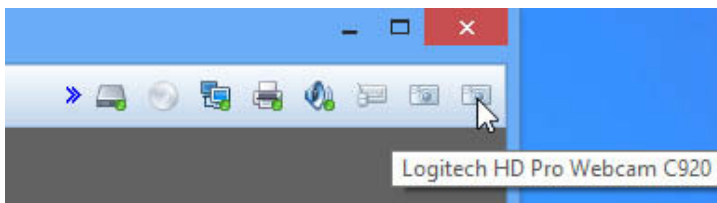


Abbildung 12: Verbinden von Geräten mit VMware I

Durch Überfahren auf die Symbole zeigt VMware die Gerätebezeichnung. Auf das richtige Gerät (Webcam oder Spiegelreflexkamera) klickt man mit der rechten Maustaste und wählt den Menüpunkt „Connect (Disconnect from host)“

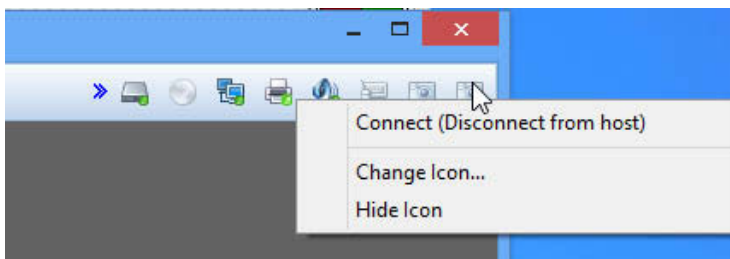


Abbildung 13: Verbinden von Geräten mit VMware II

Bei einer Spiegelreflexkamera bekommt diese das gleiche Kamera-Icon:

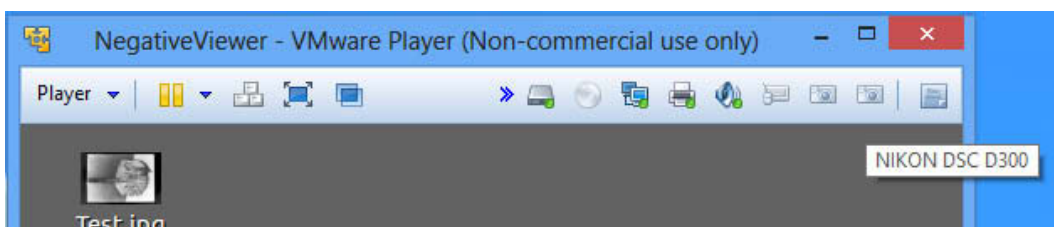


Abbildung 14: Verbinden von Geräten mit VMware III

Die Verbindung funktioniert identisch (rechte Maustaste → connect).

Anschließend kann das Programm „NegativeViewer“ von dem Desktop gestartet werden. Im folgenden Dialogfeld muss man auswählen, ob eine Webcam oder Spiegelreflexkamera angeschlossen ist.

5.9. Installation auf ein eigenes Betriebssystem

Siehe Anhang D.

6. Auswertung und Test der Kamerasysteme

Dieses Kapitel stellt die eingesetzten und getesteten Kameras und ihre Eigenschaften vor. Es handelt sich um die Webcam Logitech C920 und die Spiegelreflexkamera Nikon D300 bzw. Canon EOS 5D Mark II. Aufgrund der eingesetzten Bibliotheken funktioniert die Software vermutlich auch mit anderen Kameras.

6.1. Webcam Logitech C920

Bei dieser Webcam handelt es sich um eine Full-HD Kamera, d.h. sie kann eine Auflösung von 1920x1080 Pixeln liefern. Sie hat einen USB 2.0 Anschluss. Da die Datenrate von 100-200 MB/s für USB 2.0 zu langsam ist, besitzt sie einen eingebauten H.264 Encoder, damit die Kamera auch bei Full HD 30 Frames pro Sekunde liefert. Der Encoder lässt sich auch deaktivieren, dann schafft die Kamera aber nur noch 5 Frames/s. Als Optik besitzt die Webcam eine Zeiss Optik mit einem 20-stufigen Autofokus.

Insgesamt lässt sich über die Kamera sagen, dass sie eine für eine Webcam sehr gute Bildqualität liefert und eine erstaunliche Makrofunktionalität besitzt. Man kann bis ca. 3 cm an das Objekt herangehen. Als Nachteil ist zu sagen, dass das Objektiv besonders im Makro-modus an den Rändern eine starke Unschärfe besitzt. Am meisten stört jedoch der langsame, träge Autofokus. Daher wurde in das Programm eine Option eingebaut, mit der sich der Autofokus deaktivieren und manuell steuern lässt. Die Belichtungsmessung und Weißabgleich funktionieren gut.

Insgesamt lässt sich sagen, dass sich die Kamera für die Sichtstation eignet. Mit 80 Euro ist sie sehr preisgünstig und die abgespeicherten Bilder reichen für eine Metadatenerfassung völlig aus.

6.2. Camcorder Canon HV30

Mit dem Camcorder wurde nur kurz experimentiert. Da keines der HDMI Interfaces funktionsfähig war, konnte das Bild nicht im Rechner dargestellt werden.

Die Makrofähigkeit der Kamera ist sehr gut, auch hier kann man bis 2-3 cm an das Objekt herangehen. Das Objektiv wird an den Rändern nur wenig unschärfer.

6.3. Spiegelreflexkamera Nikon D300

Als Beispiel zur Steuerung einer Spiegelreflexkamera wurde eine Nikon D300 verwendet. Sie wird per USB 2.0 angeschlossen. Über USB liefert das Liveview Bild eine Größe von 640 x 426 Pixeln. Die Kamera lässt sich komplett vom Computer steuern. Daher wurde eine Funktion eingebaut, mit der man per Software den Auslöser betätigen kann. Nach dem Fotografieren wird das Bild direkt in das Sichtprogramm geladen, ohne dass man die Bilder manuell kopieren muss. Damit ist die volle Kameraauflösung nutzbar. Zusammen mit einer Makrooptik ergeben sich mit der Spiegelreflexkamera hervorragende Digitalisate der Originale.

Neuere Modelle von Nikon und Canon liefern per Liveview etwa 1000x700 Pixel und eignen sich daher für das schnelle Sichten mit dem Livebild besser.

Die Kamerasteuerung funktioniert aber nur mit höherwertigen Modellen von Nikon oder Canon wie z.B. Nikon D300, D600, D700, D800, D3, D4 und wahrscheinlich auch mit D5100 und D7000. Bei Canon sind das die Modelle 50D, 60D, 6D, 5D und 1D und wahrscheinlich auch 600/650D.

Die aufgelisteten Modelle konnten mangels Verfügbarkeit nicht alle ausprobiert werden und müssen beim Einsatz entsprechend getestet werden. Nicht jede Kamera lässt sich per USB steuern.

Neben der Nikon D300 wurde auch eine Canon EOS 5D Mark II getestet, mit dieser das Programm auch zusammenarbeitet. Ganz wichtig ist, dass die Kameras wie im Anhang A beschrieben entsprechend eingestellt sind. Die Beschreibung sollte sich auch für andere Modelle der Hersteller übertragen lassen.

7. Fazit und Ausblick

Das Projekt konnte erfolgreich umgesetzt werden. Der ursprüngliche Plan, die HDMI Schnittstelle zu verwenden, musste jedoch aufgegeben werden. Die Gründe dafür waren zwei defekte HDMI Interfaces, für die in der kurzen Zeit kein Ersatz beschafft werden konnte.

Daher wurde wie bereits erwähnt auf eine USB Webcam und Spiegelreflexkamera zurückgegriffen.

Die entwickelte Software lässt sich auf mehrere Themenfelder weiterentwickeln. Als weitere Eingabequelle wäre eine HDMI Schnittstelle sinnvoll. Mit dieser Option ergibt sich für die Sichtungsstation volle Flexibilität in Bezug auf die eingesetzte Kamera.

Sinnvoll ist außerdem ein integrierter HDR Modus für die Spiegelreflexkamera. Mit dem High Dynamic Range Modus werden mehrere Fotos mit unterschiedlichen Belichtungseinstellungen angefertigt und zu einem Bild verrechnet, welches einen sehr hohen Kontrastumfang in allen Tonwertbereichen besitzt.

Weitere Funktionen umfassen die optionale Eingabe von Metadaten in den IPTC Feldern und teilautomatisiertes Ausfüllen der Felder (technische Metadaten). Für eine kommerzielle Nutzung sind Colormanagement und Farbprofile ein wichtiges Thema. Bis jetzt werden keine Profile beim Abspeichern eines Bildes eingebunden.

Bei den Durchlichtmaterialien könnte man sich noch eine Funktion zum Entfernen der Orangemaskierung der Farbnegative vorstellen.

Im Bereich automatisierter Bildanalyse und Metadatenerfassung ergeben sich weitere spannende Anwendungsfelder. So kann etwa eine Gesichtserkennung hilfreich sein. Dies würde beim Erfassen von Aufnahmen mit Personen die Zuordnung erleichtern. Auch eine automatisierte Identifizierung von Gegenständen oder Art der Aufnahme (Landschaft, Person usw.) kann hilfreich sein.

Als weitere Vision kann die vollautomatisierte Digitalisierung von Negativen und Positiven genannt werden. Dabei kann auf Pick and Place Roboter zurückgegriffen werden. Je nach Modell verfügen diese über eine Ansaugvorrichtung, mit der sich Objekte aufnehmen und absetzen lassen. Somit steuert die Software den Roboter, nimmt von einem Stapel die Negative auf, platziert dieses auf dem Leuchtpult und löst die Kamera aus. Anschließend wird das Objekt auf einem anderen Stapel abgelegt.

Damit das Projekt einen großen Personenkreis erreicht und vielen Nutzern zur Verfügung steht, wird das Projekt als Open-Source veröffentlicht⁸. Es ist wünschenswert, dass noch weitere Personen das Projekt nutzen und weiter ausbauen.

⁸ <https://github.com/JohannesBiermann/NegativeViewer>

Anhang A: Stichwortartige Beschreibung der Kameraeinstellungen

Canon:

Auf den Knopf AF-Drive gehen und mit dem Wahlrad auf den Modus „One Shot“ stellen. Dann mit dem Knopf für den Liveview Modus in den Liveviewmodus wechseln (der Spiegel klappt hoch und das Sucherbild ist auf dem Bildschirm zu sehen). Auch im Liveviewmodus den Knopf AF-Drive drücken und mit dem Wahlrad auf AF Quick stellen.

Jetzt in dem Kameramenü (Knopf MENU erste rote Registerkarte Aufnahme 1) bei Qualität das RAW-Format deaktivieren und die gewünschte JPEG Größe wählen (RAW wird nicht unterstützt).

Nikon

Zunächst muss die Kamera als PTP USB Gerät eingestellt sein und nicht als Massenspeicher. Dazu geht man mit dem Knopf MENU in das Kameramenü und wählt die Registerkarte System aus (die Karte mit dem Schraubenschlüssel). Hier geht man mit dem Steuerkreuz zu dem Punkt USB und drückt OK. Jetzt wählt man MTP/PTP mit dem Steuerkreuz aus und bestätigt mit OK. Nun den Fokushebel vorne rechts am Objektivbajonett auf den Modus C (kontinuierlicher Autofokus) stellen. Bei manchen Nikon-Modellen ist diese Einstellung im Kameramenü. Hier auf kontinuierlichen Fokus umstellen.

Den Button QUAL gedrückt halten und mit den vorderen bzw. hinteren Wahlrad auf JPEG Large (L) und Fine stellen. TIFF Fine funktioniert auch. Bei manchen Nikon-Modellen ist diese Einstellung im Kameramenü zu finden. Der RAW Modus muss deaktiviert sein (RAW wird nicht unterstützt).

Anhang B: Fragen und Antworten zu Problemen mit der Kamera und dem Programm

In diesem Abschnitt sind einige häufige Probleme erklärt, die im Betrieb mit der Kamera auftreten können.

Webcam

Es ist eine Webcam angeschlossen und das Bild friert ein und auf dem Terminal erscheint die Meldung „select timeout“.

Dies ist ein Problem mit VMware. VMware kann die Signale der Kamera nicht schnell genug verarbeiten. Auf dem Computer sollten alle rechenintensiven Programme geschlossen werden. Was meistens hilft, ist die Verbindung zur Kamera zu trennen (USB Stecker abziehen und einstecken) und das Gerät wieder VMware zuweisen. Sollte dies nicht helfen, einfach die virtuelle Maschine neu starten. Dies ist ausdrücklich ein Problem mit VMware.

Spiegelreflexkamera

Es ist eine Spiegelreflexkamera angeschlossen und das Programm erkennt die Kamera nicht.

Hierzu erst überprüfen, ob das USB Gerät VMware zugewiesen wurde. Außerdem muss die Kamera auf den PTP Modus gestellt werden. Dies ist im Anhang A erklärt. Die Kamera muss sich per USB steuern lassen und Liveview besitzen. Dies ist nur bei neueren Modellen und bei (semi-) professionellen Modellen von Canon und Nikon der Fall.

Nikon: Nach Drücken des Buttons „Take Photo“ oder „Autofokus“ hängt das Programm.

Der Autofokusmodus der Kamera ist falsch eingestellt. Die Kamera muss im „Continuous“ Mode (C Mode, nicht S Mode) betrieben werden. Dies wird im Anhang A erklärt.

Nikon: Nach Drücken des Buttons „Take Photo“ scheint das Programm zu hängen und die LED für den Speicherkartenzugriff leuchtet permanent.

Dieses Verhalten wurde bei Verwendung einer Nikon D3 beobachtet. Hier scheint ein Problem mit GPhoto2 oder dem PTP Protokoll vorzuliegen. Die ganze Speicherkarte wird gescannt, was bei einer vollen Speicherkarte lange dauern kann. Abhilfe: Die Speicherkarte löschen oder die Speicherkarte entfernen.

Canon: Der Autofokus funktioniert nicht.

Die Kameraeinstellungen wie im Anhang A beschrieben überprüfen. Der Autofokus muss auf Single Shot und im Liveview Modus auf AF Quick stehen.

Anhang C: Bibliotheken und Entwicklungsumgebung installieren

Zunächst wird das QT4 Framework benötigt. Dies kann unter Debian/Ubuntu mit dem Befehl

```
apt-get install qt-sdk build-essential
```

installiert werden. Für dieses Projekt wurde QT 4.8 verwendet.

GPhoto2 installieren

Das Projekt verwendet GPhoto 2.5.x. Diese Version ist bei den Linux Distributionen noch nicht vorhanden, deshalb muss es vom Quellcode kompiliert werden. Dazu werden die Dateien gphoto2-2.5.x.tar.gz und libgphoto2-2.5.x.tar.bz2 von <http://www.gphoto.org/> heruntergeladen. X ist dabei die Version. Den Dateinamen bitte an die aktuelle Version anpassen anpassen.

Nun wird das Terminal gestartet und in das Downloadverzeichnis gewechselt:

```
cd /home/user/Downloads
```

und die Dateien entpackt:

```
tar xzf gphoto2-2.5.x.tar.gz
tar jxf libgphoto2-2.5.x.tar.bz2
```

Zuerst muss die Bibliothek kompiliert werden:

```
cd libgphoto2-2.5.x
```

Weitere Abhängigkeiten installieren

```
sudo aptitude install libpopt-dev libusb-dev libtool libltdl-dev libaa1-dev
libjpeg8-dev libreadline-dev libcdk5-dev libpopt-dev libexif-dev libgd2-xpm-dev
libtiff-dev
./configure
make all
sudo make install all
sudo ldconfig
```

Jetzt muss gphoto2 installiert werden. Gphoto2 ist der Kommandozeilenclient. Dieser kann nützlich sein, um die Kamera per Kommandozeile zu steuern.

```
cd ..
cd gphoto2-2.5.x
./configure
make
sudo make install
```

OpenCV 2.4.x installieren

Wie bei GPhoto2 wurde für dieses Projekt die neueste OpenCV Version verwendet. Bei den aktuellen Linux-Distributionen (stand Februar 2013) ist meist nur die Version 2.3 enthalten.

Zunächst müssen folgende Abhängigkeiten installiert werden:

```
mkdir build
cd build
sudo apt-get install build-essential checkinstall cmake pkg-config yasm libtiff4-dev
libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev
libdc1394-22-dev libxine-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev
libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev
libxvidcore-dev
```

Hinweis: Dieser Befehl muss auf als ein Befehl auf einer Zeile ausgeführt werden.

x264 installieren:

```
wget ftp://ftp.videolan.org/pub/videolan/x264/snapshots/last_stable_x264.tar.bz2
tar jxf last_stable_x264.tar.bz2
cd x264-snapshot-VERSIONBITTEERSETZEN
./configure --enable-shared --enable-pic
make
sudo make install
cd ..
```

v4l installieren

```
wget http://www.linuxtv.org/downloads/v4l-utils/v4l-utils-0.9.3.tar.bz2
```

Hinweis: Ggf. die neueste Version von <http://www.linuxtv.org/downloads/> herunterladen.

```
tar jxf v4l-utils-0.9.3.tar.bz2
cd v4l-utils-0.9.3
./configure
make
sudo make install
cd ..
```

ffmpeg installieren

```
wget http://ffmpeg.org/releases/ffmpeg-1.1.2.tar.gz
```

Hinweis: Ggf. die neueste Version von <http://ffmpeg.org/releases> herunterladen.

```
tar zxf ffmpeg-1.1.2.tar.gz
cd ffmpeg-1.1.2
```

```
./configure --enable-gpl --enable-libfaac --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libtheora --enable-libvorbis --enable-libx264 --enable-libxvid --enable-nonfree --enable-postproc --enable-version3 --enable-x11grab --enable-shared
```

Hinweis: Dieser Befehl ist auf einer Zeile auszuführen

```
make  
sudo make install  
cd ..
```

OpenCV kompilieren

Jetzt kann die neueste Version von OpenCV installiert werden. Diese kann mit dem Webbrowser von

<http://sourceforge.net/projects/opencvlibrary/files/latest/download?source=files> bzw.

<http://opencv.org/downloads.html> in das Verzeichnis „build“ heruntergeladen werden.

In dieser Anleitung ist die aktuelle Version 2.4.3. Bitte die Version in den Kommandos anpassen, wenn eine neuere Version zum Einsatz kommt.

```
tar jxf OpenCV-2.4.3.tar.bz2  
cd OpenCV-2.4.3  
mkdir compile  
cd compile  
cmake -D CMAKE_BUILD_TYPE=RELEASE ..  
make  
sudo make install  
sudo ldconfig
```

OpenCV und GPhoto2 sind nun installiert. Das Programm lässt sich nun starten bzw. mit der QT Creator Entwicklungsumgebung weiter entwickeln.

Anhang D: Installation auf ein eigenes Betriebssystem

Soll die Software auf einem eigenen Betriebssystem installiert werden, so sind die Abhängigkeiten GPhoto2, QT4 und OpenCV zu installieren. Zu dem momentanen Zeitpunkt bringen die Distributionen noch nicht die passenden Versionen im Repository mit, daher müssen diese wie in Anhang C beschrieben, kompiliert werden.

Die Installation selbst ist sehr einfach. Die Binärdatei muss an eine geeignete Stelle kopiert werden und kann mit einem Doppelklick ausgeführt werden.

Sollten die passenden Versionen OpenCV 2.4.x und Gphoto2 2.5.x im Repository verfügbar sein, so genügt

```
sudo aptitude install libopencv-video2.4  
sudo aptitude install gphoto2 libgphoto2-2  
sudo aptitude install libqt4-gui
```

Zum jetzigen Zeitpunkt existieren diese Pakete in den benötigten Versionen noch nicht im Ubuntu 12.10 Repository. Die oberen Befehle können daher leicht variieren.

Anhang E: Abbildungsverzeichnis

Abbildung 1: Repräsentativ mit demontierter Kameraaufnahme	3
Abbildung 2: Halterung für die Repräsentativkamera (Ansicht frontal und von oben).....	3
Abbildung 3: Erster Versuchsaufbau mit Stativkopf und Leuchtpult	5
Abbildung 4: OpenCV Bildmatrix.....	6
Abbildung 5: Automatische Tonwertkorrektur (Quelle: http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html).....	9
Abbildung 6: Automatisches Freistellen Ausgangsbild	11
Abbildung 7: Automatisches Freistellen invertiertes Graustufenbild (weißer Rand entfernen).....	11
Abbildung 8: Automatisches Freistellen Binärbild mit markiertem Ausschneidebereich.....	11
Abbildung 9: Automatisches Freistellen fertiges Ergebnis	12
Abbildung 10: Bedienoberfläche des Programms	13
Abbildung 11: Optionen bei Betrieb mit einer Webkamera	14
Abbildung 12: Verbinden von Geräten mit VMware I.....	16
Abbildung 13: Verbinden von Geräten mit VMware II.....	16
Abbildung 14: Verbinden von Geräten mit VMware III.....	16

Anhang F: Literaturverzeichnis

Fanning, David W.: Image Sharpening with a Laplacian Kernel,

http://www.idlcoyote.com/ip_tips/sharpen.html, abgerufen 16.02.2013

Loch, G. Francis: Image Processing Algorithms Part 7: Colour Inversion And Solarisation,

<http://www.dfstudios.co.uk/articles/image-processing-algorithms-part-7-colour-inversion-and-solarisation>, abgerufen 25.01.2013

N.N.: Histogram Equalization, http://math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf,
abgerufen 25.01.2013

OpenCV Dev Team: Mat - The Basic Image Container,

http://docs.opencv.org/doc/tutorials/core/mat_the_basic_image_container/mat_the_basic_image_container.html#matthebasicimagecontainer, abgerufen 16.02.2013

OpenCV Dev Team: Miscellaneous Image Transformations,

http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor,
abgerufen 25.01.2013

Wikipedia, The Free Encyclopedia: YUV,

<http://en.wikipedia.org/w/index.php?title=Special:Cite&page=YUV&id=533236206>,
abgerufen 25.01.2013