

Abbildung 1: Beispiel einer kolorierten Fotografie (Image Colorization). Drei einzelne Farbkanäle (rot, grün, blau) werden zu einem Farbbild kombiniert.

## CV/task1a—Image Colorization

Ziel dieser Aufgabenstellung ist es, grundlegende Funktionalitäten sowie einen Teil des erweiterten Funktionsumfangs der OpenCV-Library kennenzulernen. Dafür soll Schritt für Schritt ein automatisiertes *Image Colorization*-Programm entwickelt werden. Dabei soll aus drei Einzelbildern (Farbkanäle rot, grün und blau) ein koloriertes Farbbild erzeugt werden. Ein Beispiel finden Sie in Abbildung 1.

Für die ausreichende Darstellung eines Farbbildes sind zumindest drei Farbinformationen pro Pixel notwendig (z. B. RGB, Lab, HSV, ...). Diese werden als Koordinaten in dem jeweiligen Farbraum interpretiert und definieren dadurch eindeutig die Farbe des Pixels. Eine gängige Form der Darstellung bietet der RGB-Farbraum mit seinen drei Koordinaten für den (roten) R-, (grünen) G- und (blauen) B-Kanal. Dieses Beispiel befasst sich mit der Rekonstruktion eines Farbbildes aus drei separaten, nacheinander aufgenommenen und teils verschobenen Intensitätsbildern. Diese wurden nach dem Verfahren von Sergei Michailowitch Prokudin-Gorski<sup>1</sup> erstellt. Dabei wurden in kurzem zeitlichen Abstand drei Intensitätsbilder durch jeweils einen roten, blauen und einen grünen Farbfilter aufgenommen. Um zu einem späteren Zeitpunkt ein Farbbild wiedergeben zu können, wurden drei Projektoren – einer mit einem roten, einer mit einem grünen und einer mit einem blauen Farbfilter – verwendet, mit denen die Einzelbilder übereinander projiziert wurden.

---

<sup>1</sup>[https://de.wikipedia.org/wiki/Sergei\\_Michailowitsch\\_Prokudin-Gorski#Fotografisches\\_Verfahren](https://de.wikipedia.org/wiki/Sergei_Michailowitsch_Prokudin-Gorski#Fotografisches_Verfahren)

# 1 Aufgaben

Die benötigten Parameter der für diese Aufgabe vorgeschlagenen Funktionen aus OpenCV sind in den jeweiligen JSON-Dokumenten hinterlegt. Nicht vorhandene, in der OpenCV-Dokumentation jedoch angegebene Parameter können für dieses Beispiel vernachlässigt werden, indem die Standardwerte verwendet werden.

Diese Abgabe gliedert sich in folgende Teilaufgaben, die aufeinander aufbauen:

- Kantendetektion
- Berechnung der Verschiebungen und Translation der Bilder
- Kombinieren der Farbkanäle
- Zuschneiden des Resultates

## 1.1 Kantendetektion (1.5 Punkte)



(a) Eingabe (G-Kanal).



(b) img\_g\_edge (Kanten G-Kanal).

Abbildung 2: Eingabe und Ausgabe der Kantendetektion.

Die Kantenerkennung soll mit Hilfe der von OpenCV zur Verfügung gestellten Funktion der ‘Sobel-Derivates’ durchgeführt werden. Die hierfür anwendbare OpenCV Funktion Sobel detektiert gerichtete Änderungen unter Zuhilfenahme von Ableitungen. Ein großer Gradientenwert eines Bildes deutet auf eine Änderung im Bild hin (z. B. Kanten, Farbübergänge). Für die Ableitung in x-Richtung  $G_x$  wird das Bild  $I$  mit dem Kern

$S_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$  gefaltet. Analog wird für die Ableitung in y-Richtung  $G_y$  das Bild  $I$

mit dem Kern  $S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$  gefaltet, sodass sich die Formeln

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \quad (1)$$

und

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \quad (2)$$

ergeben.  $*$  bezeichnet dabei die Faltungsoperation. Bestimmen Sie zuerst die jeweils *erste* Ableitung in die Richtungen x und y. Wählen Sie für den Parameter der die Bit-Tiefe des Output-Bildes beschreibt den Typ CV\_32F. Kombinieren Sie danach die beiden Gradientenbilder gemäß der Formel

$$G = \sqrt{G_x^2 + G_y^2}, \quad (3)$$

wobei  $G$  das Gesamtgradientenbild bezeichnet.

Anschließend soll das Resultat wieder in ein Intensitätsbild konvertiert und eine geeignete Skalierung verwendet werden. Nutzen Sie hierfür die Methode Mat::convertTo(..) und skalieren Sie die Pixelwerte sodass die Werte von  $px_{min}$  – welcher hier dem niedrigsten im Gradientenbild vorkommenden Pixelwert entspricht – bis  $px_{max}$  – welcher dem größten Pixelwert entspricht – in das Intervall [0, 255] abgebildet werden. Um ein besseres Ergebnis der Kantendetektion zu erhalten und fälschlich detektierte Kanten zu eliminieren, wird nun Thresholding auf das Bild angewandt, sodass alle Werte, die kleiner oder gleich dem Parameter  $\tau$  sind, 0 gesetzt werden, und jene die größer als der Parameter sind, werden auf 255 gesetzt. Das binäre Bild  $E$

$$E(x, y) = \begin{cases} 255, & \text{wenn } G(x, y) > \tau \\ 0, & \text{sonst} \end{cases} \quad (4)$$

enthält somit nur noch die Kanteninformation. Entnehmen Sie den Wert von  $\tau$  der Konfigurationsdatei unter dem Namen edge\_threshold. Die Resultate  $E$  dieses Teilschrittes sollen in den Matrizen img\_r\_edge, img\_g\_edge und img\_b\_edge des jeweiligen Farbkanals gespeichert und werden.

Abbildung 2 zeigt beispielhaft das Ergebnis der Kantendetektion.

## 1.2 Berechnung der Verschiebungen und Translation der Bilder (2.5 Punkte)

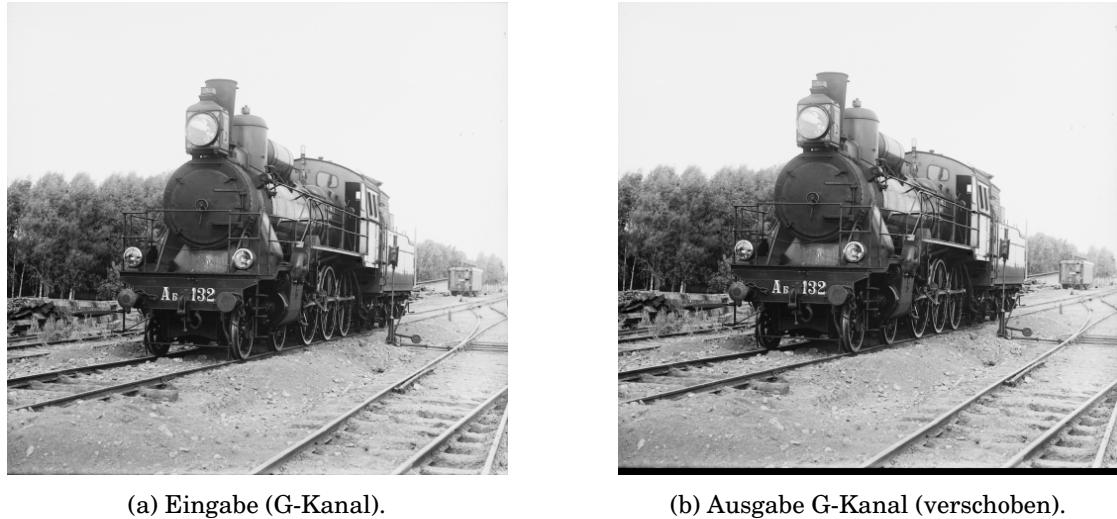


Abbildung 3: Eingabe und Ausgabe der Verschiebungsbestimmung und Translation der Bilder.

Da die unterschiedlichen Eingabebilder leichte Abweichungen bezüglich der Position der in ihnen vorkommenden Objekte aufweisen, müssen in diesem Schritt die Verschiebungen (jeweils in  $x$ - und  $y$ -Richtung) zwischen den Bildern bestimmt werden. Dabei wird das Eingabebild «01.png» – der rote Farbkanal – als Referenz gewählt. Hierfür wird innerhalb eines durch den Parameter `match_window_size` definierten Bereiches die optimale Verschiebung berechnet. Um diese optimale Verschiebung zu bestimmen, werden die Kantenbilder der beiden übereinander zu legenden Kanäle jeweils von  $\min_{i,j} = -\text{match\_window\_size}/2$  bis inklusive  $\max_{i,j} = \text{match\_window\_size}/2$  verschoben und eine Punktzahl für die jeweilige Verschiebung bestimmt. Diese Punktzahl ist dabei definiert als die *Summe der Kantenpixel im Referenzkanal, welche in dem transformierten Bild ebenfalls auf einem Kantenpixel liegen*

$$score = \operatorname{argmax}_{i,j} \sum_{x,y} \frac{E(x,y)}{255} \cdot \frac{E'(x+i,y+j)}{255} \quad i,j \in [\min_{i,j}, \dots, \max_{i,j}], \quad (5)$$

wobei  $E$  das Kantenbild des Referenzkanals und  $E'$  eines der Kantenbilder der Kanäle B und G definiert, und  $x$  und  $y$  für alle Pixel in Bild stehen.

Wie in Abbildung 4 zu sehen ist, wird das zu verschiebende Bild (gelb) zunächst um die maximale Verschiebung ( $i = \min_{i,j}$  und  $j = \min_{i,j}$ ) verschoben. Danach wird die Punktzahl für diese Verschiebung bestimmt, mit der Punktzahl der bisher besten Verschiebung

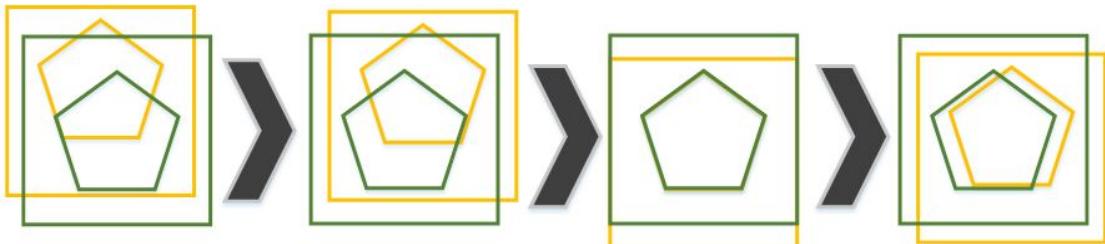


Abbildung 4: Kantenabgleich (schematische Darstellung). Der dritte Schritt zeigt die optimale Translation.

verglichen (und evtl. gespeichert), danach wird die x- oder y-Verschiebung angepasst und wiederum die Punktzahl bestimmt, bis alle möglichen Verschiebungs-Kombinationen berechnet und die optimale Verschiebung bestimmt wurde. In der schematischen Darstellung in Abbildung 4 wäre das dritte Teilergebnis optimal, da alle Kantenpixel im Referenzkanal (grün) auf einem Kantenpixel im zu verschiebenden Bild (gelb) liegen.

Beispielsweise wird während des Kantenabgleich-Vorganges ein Pixel mit den Koordinaten (50,50) im Ursprungsbild bei einer `match_window_size` von 20 Pixel, einmal auf jedem Pixel von (40,40) bis (60,60) liegen.

Bestimmen Sie die optimalen Verschiebungen und transformieren Sie anschließend die Bilder `img_g` und `img_b` indem Sie die Bilder entlang der x- und y-Achse gemäß der von Ihnen bestimmten optimalen Verschiebungen verschieben. Speichern Sie Ihre Ergebnisse in den Matrizen `img_g_aligned` bzw. `img_b_aligned`.<sup>2</sup> Abbildung 3 zeigt das Ergebnis des Kantenableichs zur Bestimmung der optimalen Translation.

### 1.3 Kombinieren der Farbkanäle (1 Punkt)

Nachdem Sie in den bisherigen Abschnitten die Kanäle so transformiert haben, dass sie nun bestmöglich übereinander liegen, sollen Sie in einem weiteren Schritt die Einzelbilder zu einem Gesamtbild mit drei Kanälen (RGB) kombinieren.

Bestimmen Sie also für jedes Pixel seinen (R,G,B)-Wert aus dem Referenzkanal bzw. den beiden im vorigen Schritt transformierten Kanälen. Achten Sie weiterhin auf die Korrektheit der Indizes – liegt ein Index außerhalb des gültigen Bereiches für einen Kanal, so setzen Sie den Kanal des entsprechenden Pixels auf 0. Wiederholen Sie dies für alle weiteren Pixel und speichern Sie Ihr Ergebnis in der Matrix `out`.<sup>3</sup>

---

<sup>2</sup>Die Abmessungen der transformierten Bilder sollen mit jenen der Eingabebilder übereinstimmen.

<sup>3</sup>Für diese Übung soll die Matrix `out` dieselben Abmessungen wie der Referenzkanal (`img_r`) haben.

## 1.4 Zuschneiden des Resultates (1 Punkt)



(a) Kombinierte Bildkanäle.



(b) Ausgabe nach dem Zuschneiden.

Abbildung 5: Eingabe und Resultat des Zuschneide-Prozesses.

Da im vorangegangen Schritt die Einzelkanäle kombiniert, nicht aber überprüft wurde wo der Bild-Bereich endet in dem alle drei Kanäle gültige Werte haben, kann es zu Rändern kommen, an denen ein oder mehrere Kanäle undefiniert sind, wie in Abbildung 5 zu sehen. Um diese zu beseitigen, verkleinert man den Bildbereich. Dafür muss zuerst der Schnittbereich – also jener Bereich, der ausschließlich gültige Werte enthält – bestimmt werden. Er kann anhand der Abmessungen der Bilder – `img_r`, `img_g_aligned`, `img_b_aligned` – und der Verschiebungen zwischen den Kanälen R und G sowie, R und B bestimmt werden.

Abbildung 6 zeigt schematisch die einzelnen Kanäle (R (rot), G (grün) und B (blau)), sowie den dazugehörigen Schnittbereich (schraffiert). Schneiden Sie diesen Bereich aus und speichern Sie ihn in `out_cropped`.

### Hilfreiche OpenCV-Befehle:

- `cv::Sobel(...)`
- `cv::minMaxLoc(...)`
- `cv::Mat::convertTo(...)`
- `cv::threshold(...)`
- `cv::Rect(...)`

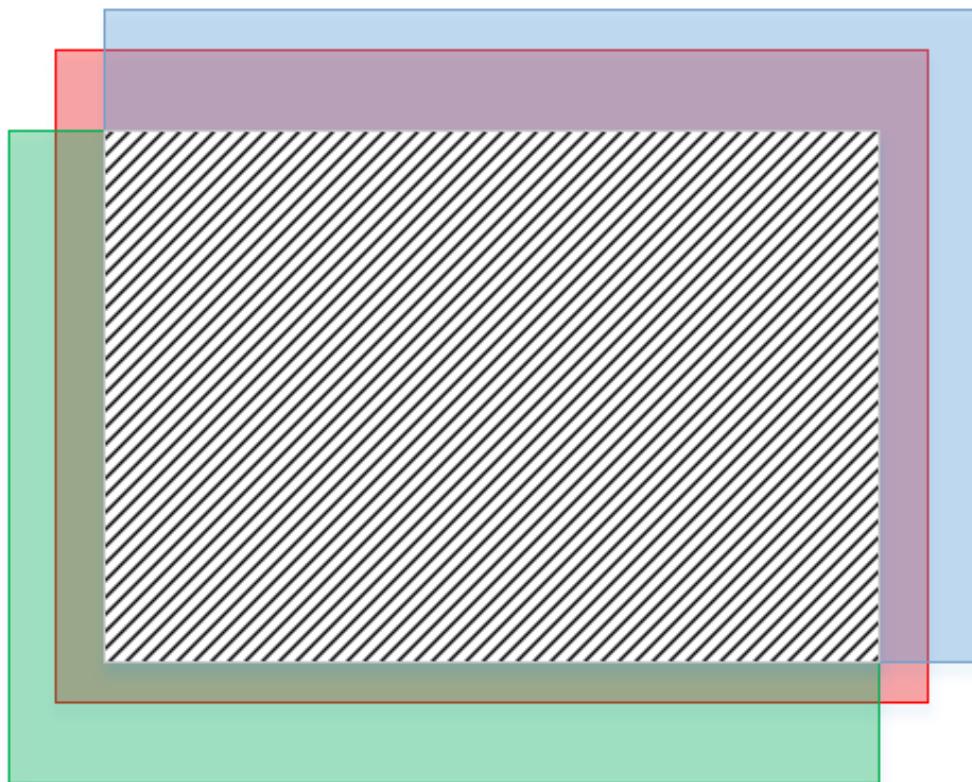


Abbildung 6: Schnittbereich als Überlagerung der einzelnen Bild-Kanäle. Der schraffierte Bereich soll ausgeschnitten werden.

- `cv::Mat::copyTo(...)`

## 2 Ein- und Ausgabeparameter

Die folgenden Parameter in der Konfigurationsdatei bestimmen das Verhalten des Programms (Anwendungsbeispiele finden Sie in den Dateien «train.json», «cathedral.json» und «emir.json»):

- Kantendetektion Schwellwert  $\tau$ : `edge_threshold`
- Maximale Verschiebung: `match_window_size`
- Eingabe-Pfad: `input_path`

- Eingabe (B-Kanal): img\_b
- Eingabe (G-Kanal): img\_g
- Eingabe (R-Kanal): img\_r
- Ausgabe Kantendetektion (B-Kanal): img\_b\_edge
- Ausgabe Kantendetektion (G-Kanal): img\_g\_edge
- Ausgabe Kantendetektion (R-Kanal): img\_r\_edge
- Ausgabe Translation (G-Kanal): img\_g\_aligned
- Ausgabe Translation (B-Kanal): img\_b\_aligned
- Ausgabe kombinierte Farbkanäle: out
- Ausgabe (zugeschnitten): out\_cropped

### 3 Programmgerüst

Die folgende Funktionalität ist in dem vom ICG zur Verfügung gestellten Programmgerüst bereits implementiert und muss von Ihnen nicht selbst programmiert werden:

- Die JSON-Konfigurationsdatei wird vom Programmgerüst gelesen.
- Lesen der Eingabebilder.
- Lesen des edge\_threshold.
- Lesen der match\_window\_size.
- Schreiben der Ausgabebilder.
- Ihre Implementierung ist in main.cpp durchzuführen.

**Diese Aufgabe ist mit Hilfe von OpenCV<sup>4</sup> zu implementieren. Nutzen Sie die Funktionen, die Ihnen OpenCV zur Verfügung stellt und achten Sie auf die unterschiedlichen Parameter und Bildtypen! Beachten Sie auch, dass die An-**

---

<sup>4</sup><http://opencv.org>

**dung der Farbkanäle in OpenCV ( $[B, G, R]$  und nicht  $[R, G, B]$ ) ist. Das C++ Cheatsheet<sup>5</sup> könnte bei dieser einfachen Aufgabe sehr hilfreich sein.**

---

<sup>5</sup>[http://docs.opencv.org/2.4/opencv\\_cheatsheet.pdf](http://docs.opencv.org/2.4/opencv_cheatsheet.pdf)

## 4 Abgabe

Die Aufgaben bestehen jeweils aus mehreren Schritten, die zum Teil aufeinander aufbauen, jedoch unabhängig voneinander beurteilt werden. Dadurch ist einerseits eine objektive Beurteilung sichergestellt und andererseits gewährleistet, dass auch bei unvollständiger Lösung der Aufgaben Punkte erzielt werden können.

Wir weisen ausdrücklich darauf hin, dass die Übungsaufgaben von jedem Teilnehmer *eigenständig* gelöst werden müssen. Wenn Quellcode anderen Teilnehmern zugänglich gemacht wird (bewusst oder durch Vernachlässigung eines gewissen Mindestmaßes an Datensicherheit), wird das betreffende Beispiel bei allen Beteiligten mit 0 Punkten bewertet, unabhängig davon, wer den Code ursprünglich erstellt hat. Ebenso ist es nicht zulässig, Code aus dem Internet, aus Büchern oder aus anderen Quellen zu verwenden. Es erfolgt sowohl eine automatische als auch eine manuelle Überprüfung auf Plagiate.

Da die abgegebenen Programme halbautomatisch getestet werden, muss die Übergabe der Parameter mit Hilfe von entsprechenden Konfigurationsdateien genauso erfolgen wie bei den einzelnen Beispielen spezifiziert. Insbesondere ist eine interaktive Eingabe von Parametern nicht zulässig. Sollte aufgrund von Änderungen am Konfigurationssystem die Ausführung der abgegebenen Dateien mit den Testdaten fehlschlagen, wird das Beispiel mit 0 Punkten bewertet. Die Konfigurationsdateien liegen im JSON-Format vor, zu deren Auswertung steht Ihnen `rapidjson` zur Verfügung. Die Verwendung ist aus dem Programmgerüst ersichtlich.

Jede Konfigurationsdatei enthält zumindest einen Testfall und dessen Konfiguration. Es ist auch möglich, dass eine Konfigurationsdatei mehrere Testfälle enthält, um gemeinsame Parameter nicht mehrfach in verschiedenen Dateien spezifizieren zu müssen. In manchen Konfigurationsdateien finden sich auch einstellbare Parameter, die in Form eines `select` Feldes vorliegen. Diese sollen die Handhabung der Konfigurationsdateien erleichtern und ein einfaches Umschalten der Modi gewährleisten.

Es steht Ihnen frei, z.B. zu Testzwecken eigene Erweiterungen zu implementieren. Stellen Sie jedoch sicher, dass solche Erweiterungen in Ihrem abgegebenen Code deaktiviert sind, damit ein Vergleich der abgegebenen Arbeiten mit unserer Referenzimplementierung möglich ist.

Die Programmgerüste, die zur Verfügung gestellt werden, sind unmittelbar aus unserer Referenzimplementierung abgeleitet, indem nur jene Teile entfernt wurden, die dem Inhalt der Übung entsprechen. Die Verwendung dieser Gerüste ist nicht zwingend, aber Sie ersparen sich sehr viel Arbeit, wenn Sie davon Gebrauch machen.