1 Bildverarbeitung

Die Aufnahmen der Kamera werden von der Software als JPEG Dateien abgespeichert. Dies ist für eine weitere quantitative Auswertung problematisch, da die gemessenen Temperaturen in RGB-Werte umgewandelt werden. Eine wichtige Aufgabe ist es daher, die ursprünglichen Temperaturinformationen für jeden Pixel möglichst genau wiederherzustellen. Dies wird in zwei Schritten bewerkstelligt: Einerseits der Zuordnung von Farb- und Temperaturwerten mittels des im Bild inkludierten Farbbalkens, andererseits der Umwandlung aller Pixel in Temperaturen.

1.1 Farb-Temperatur Zuordnung

Jedes Bild verfügt über einen Farbbalken, welcher zum visuellen Abschätzen der Temperaturen innerhalb des Bildes dient. Der Maximal- und Minimalwert des Balkens ist jeweils mit der zugehörigen Temperatur gekennzeichnet. Mit Hilfe dieser Informationen lässt sich eine Zuordnung von Farbwert und Temperatur erzeugen. Für jeden Pixel i des Farbbalkens kann die korrespondierende Temperatur $T_{scale,i}$ mittels folgender Formel berechnet werden:

$$T_{scale,i} = \frac{i}{n} \left(T_{max} - T_{min} \right) + T_{min} \tag{1}$$

wobei n die Anzahl der Pixel des Farbbalkens representiert. Die beiden angegebenen Randtemperaturen T_{min} und T_{max} werden jeweils ganzzahlig angegebenen. Dies lässt die Frage offen, ob diese zur Präsentation auf- oder abgerundet wurden. Die Standardunsicherheit der Randtemperaturen u_{round} kann quantifiziert werden durch:

$$u_{round} = \frac{1}{2\sqrt{3}} \tag{2}$$

Folglich ergibt sich ein möglicher Fehler für die berechneten Temperaturen, welcher sich über die Fehlerfortpflanzung berechnen lässt:

$$u_{scale,i} = \sqrt{\left(\frac{i}{n} u_{round} \right)^2 + \left(\left(1 - \frac{i}{n} \right) u_{round} \right)^2} \tag{3}$$

Hier beschreibt $u_{scale,i}$ die Standardunsicherheit des berechneten Temperaturwerts der i-ten Balkenfarbe.

1.2 Farbinterpolation

Nun sind zwar die Temperaturen für alle Farben des Farbbalkens bekannt, jedoch können die Pixel des eigentlichen Bildes Farbwerte besitzen, die zwischen denen des Farbbalkens liegen. Dies Bedarf dem Einsatz von Farbinterpolation, was jedoch keine triviale Aufgabe ist. Der Farbverlauf des Farbbalkens stellt eine dreidimensionale Kurve im RGB-Raum dar, welche die Anwendung vieler der üblicher Interpolationsmethoden ausschließt. Folglich viel unsere Wahl auf Nearest-Neighbor Interpolation, da alle anderen getesteten Methoden fehlschlugen.

$$u_{ip} = \frac{T_{max} - T_{min}}{4\sqrt{3}n} \tag{4}$$

$$u_{pixel} = \sqrt{u_{scale,i}^2 + u_{ip}^2} \tag{5}$$