Automatisches Dodging von Luftbildern

Martin Drauschke

martin.drauschke@uni-bonn.de

TR-IGG-P-2006-01

14. Dezember 2006

Technischer Report Nr. 1, 2006

Professur für Photogrammetrie
Institut für Geodäsie und Geoinformation
Universität Bonn

Erhältlich unter
http://www.ipb.uni-bonn.de/~martin/

Automatisches Dodging von Luftbildern

Martin Drauschke

14. Dezember 2006

1 Konzept und Anforderungen

Das Problem stellt sich wie folgt dar: Die Luftbilder wurden mit einem Vexcel-Scanner digitalisiert und als 16-Bit-Bilder abgespeichert. Die Bilder sollen automatisch nachbereitet werden.

2 Ein- und Ausgabe

Das Programm erwartet ein Grauwertbild im `tiff`-Format mit 16 Bits pro Pixel. Zudem kann der Benutzer den Ablauf des Programms bzw. dessen Ausgabe durch die Angabe zusätzlicher Parameter direkt beeinflussen. In diesem Abschnitt werden die Eingabeparameter kurz vorgestellt und erläutert. Ein möglicher Programmaufruf könnte beispielsweise wie folgt lauten:

```
dodging.exe datei.tif -p 500 800 -s 0.5 -b 8 -i 0 -o datei2.tif -v
```

Die Reihenfolge der optionalen Parameter ist dabei egal. Die Zuordnung der Werte erfolgt durch die Schlüssel `-b`, `-i`, `-o -p` bzw. `-s`.

Das Eingabebild Das Eingabebild ist ein Grauwertbild und liegt im `tiff`-Format vor. Es ist dabei egal, ob die Datei in `tiles` oder in `strips` abgespeichert wurde. Die radiometrische Auflösung des Bildes beträgt 16 Bits. Die Bildgröße wird bei ca. 12000 x 12000 Pixeln erwartet.

Die Bilddatei muss sich nicht im Verzeichnis befinden, von dem aus das Programm aufgerufen wird. Dann wird der Verzeichnispfad beim Aufruf des Programms dem Dateinamen vorangestellt.

```
dodging.exe C:\verzeichnis\eingabe.tif
```

Der Parameter `-b` Der Parameter `-b` kann benutzt werden, wenn die radiometrische Auflösung des Ausgabebilds verändert werden soll. Hinter dem Bezeichner `-b` muss eine positive, ganze Zahl angegeben werden. Sie gibt an, mit wievielen Bits pro Pixel das Ausgabebild gespeichert werden soll. Die `default`-Belegung der Variable ist 16, d. h. das Ausgabebild wird mit derselben Graweerttiefe abgespeichert wie das Eingabebild aufweist. Alternativ kann derzeit die Ausgabedatei auch mit 8 Bits pro Pixel gespeichert werden.

Der Parameter -i Mit dem Parameter `-i` kann die Interpolationsmethode vom Benutzer ausgewählt werden. Das Programm berechnet die Transformation der Grauwerte lokal, d.h. in kleineren Patches. Wenn nicht zwischen den Patches interpoliert werden soll, dann werden die Patchgrenzen im Ausgabebild sichtbar. Hinter dem Bezeichner `-i` muss eine ganze Zahl angegeben werden. Bei 0 wird die Interpolation im Programmfluss übersprungen, bei allen anderen Werten findet eine bilineare Interpolation statt, was auch als `default`-Wert eingestellt ist.

Der Parameter -o Das Programm schreibt die Ausgabedatei `default`-mäßig in dasselbe Verzeichnis, in dem auch die Eingabedatei liegt. Wird der Name des Eingabebilds mit `C:\verzeichnis\dateiname.tiff` angegeben, dann trennt das Programm die Pfadangaben vom Dateinamen und benennt das Ausgabebild `transf_`+dateiname`.tiff` und schreibt dieses in das Verzeichnis `C:\verzeichnis\`. Die Angabe des Parameters `-o` bewirkt, dass die Ausgabedatei in ein anderes Verzeichnis und unter einem anderen Namen gespeichert wird. Dem Bezeichner muss eine Zeichenkette folgen, die aus dem neuen Verzeichnis und dem neuen Dateinamen besteht.

Der Parameter -p Der Parameter `-p` hat für die Qualität des Ausgabebilds den größten Einfluss. Hiermit kann die Patchgröße durch den Benutzer angegeben werden. Erste Tests hatten ergeben, dass eine globale Bildverbesserung nicht so gut ist, wie eine lokal arbeitende. Aus diesem Grund wird das Bild in Patches unterteilt, deren `default`-Größe 1000 x 1500 ist. Hinter dem Bezeichner `-p` müssen zwei positive, ganze Zahlen folgen. Die erste gibt die Höhe des Patches an, d.h. die Anzahl der Zeilen. Die zweite beschreibt mit der Anzahl der Spalten die Breite des Patches. Die Einteilung des Bildes in Patches erfolgt von der oberen linken Ecke, die Patches am rechten und unteren Bildrand können entsprechend kleiner sein als die anderen. Das Programm überprüft, ob die Patchgröße sinnvoll gewählt ist. So muss ein Patch mindestens 8000 Pixel enthalten. Ferner sollte das Bild in mindestens zwölf Patches eingeteilt werden. Dabei müssen sich mindestens zwei Patches nebeneinander sowie übereinander befinden.

Der Parameter -s Wenn das Bild direkt nach der Histogrammverebnung ausgegeben wird, ist der Kontrast im Bild teilweise sehr groß. Dieser kann durch ein anschließendes Stauchen des Histogramms abgemildert werden. Das Stauchen des Histogramms bewirkt, dass das Bild nicht mehr Grauwerte aus dem Intervall [0..65535] enthält, sondern aus einem kleinerem. Da der Kontrast dabei verringert wird, wird dieser Programmpunkt auch als Weichzeichnen bezeichnet. Die dabei verwendete Histogrammtransformation ist linear und wird aus den maximalen Gradienten jedes Bildpatches bzw. den Median der Gradienten abgeleitet. Damit der Stauchungsfaktor nicht zu klein wird, wird eine untere Schwelle überprüft. Dieser Wert wird zusammen mit dem Parameterkennzeichen `-s` als Dezimalzahl übergeben. Negative Werte und Werte größer als

1 werden abgelehnt. In diesen Fällen wird der betrag des Wertes verwendet bzw. der Parameter wird automatisch auf 1.0 gesetzt. Dieser Wert bewirkt keine Veränderung des Histogramms. Bei Angabe einer 0, was im Prinzip einer Stauchung des gesamten Histogramms auf einen einzigen Grauwert gleichkäme, wird die Histogrammstauchung übersprungen. Der `default`-Wert des Parameters ist 0.4.

Der Parameter `-v` Einzelne Zwischenergebnisse und Statistiken können über die Konsole ausgegeben werden. Dazu sollte der Parameter `-v` ohne zusätzliche Argumente beim Programmaufruf angegeben werden. Die Ausgabe hat dann beispielweise folgenden Inhalt:

```
Ausgabe wichtiger Programmabschnitte erfolgt.

Variablenbelegungen nach dem Parsen
  Patchsize in Zeilen: 1000
  Patchsize in Spalten: 1500
  Interpolation: 1
  Weichzeichnen: 0.4
  Bits pro Pixel: 16
  Dateiname für Ausgabe: transf_101878_16.tif

Bild 101878_16.tif wurde erfolgreich eingelesen.
  Groesse des Bildes: 109934704.
  Abmaße des Bildes (h,w): 11344 x 9691.
  Das Bild wird in 12 x 7 Patches zerteilt.

Statistik der Original-Histogramme:

  Bild hat insgesamt 65280 unbesetzte Grauwerte

  Minimale Werte der Patches
    min = 0
    max = 2314
    Mittel = 241.857
  Maximale Werte der Patches
    min = 65535
    max = 65535
    Mittel = 65535
  Differenzen zwischen maximalen und minimalen Werten der Patches
    min = 63221
    max = 65535
    Mittel = 65293.1
  Mittelwerte der Patches
    min = 16841.1
    max = 59464.8
```

```
    Mittel = 30817.5

Overflow: über allen Pixeln des Bildes:
  Anzahl im Kumulativen Histogramm: 0
  Anzahl im verebneten Histogramm:  0
  Anzahl nach Interpolation: 0

Statistik der Gradienten nach Histogrammverebnung:
  Maximale Werte der Patches
    min = 89804
    max = 92680
    Mittel = 92549.3
  Medianwerte der Patches
    min = 38741.4
    max = 64962.5
    Mittel = 48693.1

Das verebnete Histogramm wird im Mittel um folgenden Faktor gestaucht: 0.482235

Insgesamt traten 0 Under- bzw. Overflows bei der Transformation des Histogramms auf.
Das entspricht 0 Under- bzw. Overflows pro Patch.
Diese Werte wurden durch 0 bzw. 65535 ersetzt.
D.h. es gab einen Informationsverlust bei durchschnittlich 0% der Grauwerte.

Overflow: über allen Pixeln des Bildes:
  Anzahl bei Transformation des Histogramms: 0
```

Hilfestellung Durch Aufruf des Programms mit `dodging.exe -h` wird eine kleine Hilfe im Eingabefenster angezeigt. Hier werden die optionalen Parameter kurz beschrieben.

3 Programmablauf

Programmstart Das Programm überprüft zu Beginn die übergebenen Parameter. Das Eingabebild wird eingelesen, zudem wird der Speicherplatz für das Ausgabebild angelegt. Die Parameter werden initialisiert und nach dem Parsen des Programmaufrufs gemäß den Eingaben des Benutzers aktualisiert.

Globaler vs. lokaler Ansatz Weil das Verfahren mit einer globalen Histogrammverebnung in zahlreichen Bildteilen unbefriedigende Ergebnisse erzielt, wird das Bild in kleinere Patches zerlegt, in denen die Histogrammverebnung lokal durchgeführt wird. Die Anzahl der Patches wird aus der Bildgröße und den Maßen eines Patches berechnet. Die Einteilung des Bildes in diese Patches erfolgt ausgehend von der linken oberen Bildecke. Die Patches überlappen sich

nicht, die Patches am rechten bzw. am unteren Bildrand sind mitunter deutlich kleiner als die anderen.

Histogrammverebnung Um die Histogrammverebnung durchführen zu können, werden drei Histogramme benötigt, deren Speicherplatz nach der Berechnung der Anzahl der Patches angelegt wird. Pro Patch i müssen folgende drei Histogramme über 65536 Grauwerte (16 Bit) berechnet werden:

1. Histogramm H_i der Grauwerte des Eingabepatches,
2. Kumulatives Histogramm K_i der Grauwerte des Eingabepatches,
3. Abbildung A_i der transformierten Grauwerte.

Im ersten Durchlauf durch das Bild werden die Grauwerte für die Histogramme H_i gezählt. Anschließend wird das kumulative Histogramm K_i für jeden Patch berechnet:

$$K_i(0) = H_i(0) \tag{1}$$

$$K_i(j) = K_i(j-1) + H_i(j), \text{ für } j = 1, \ldots, 65535. \tag{2}$$

Aus den kumulativen Histogrammen können die Histogrammverebnungen direkt ermittelt werden. Die Abbildung der Grauwerte wird im dritten Feld abgespeichert. Dabei wird der kleinstmögliche Grauwert wieder auf den kleinstmöglichen Grauwert abgebildet:

$$A_i(0) = 0. \tag{3}$$

Der größtmögliche Grauwert wird analog auf den größtmöglichen Grauwert transformiert:

$$A_i(65535) = 65535. \tag{4}$$

Die Abbildungen für die Grauwerte $j \in (0, 65535)$ ergeben sich nach folgender Formel:

$$A_i(j) = \frac{65535}{n - K_i(0)} \left(K_i(j) - K_i(0)\right), \tag{5}$$

wobei n die Bildgröße, d.h. die Anzahl der Pixel im Bild ist.

Behauptung: Der Wertebereich einer Histogrammverebnung ist [0..65535], d.h. die Histogrammverebnung führt zu keinem Under- bzw. Overflow von Grauwerten.

Beweis: Die Gleichung (**??**) zeigt eine Funktion über der Variable j, so dass sie auch als Funktion $f(j)$ gelesen werden kann,

$$f(j) = c_1 \left(g(j) - c_2\right), \tag{6}$$

d.h. die Funktionswerte hängen nur von den Einträgen des Kumulativen Histogramms und von 2 Konstanten, c_1 und c_2 ab. Berechnet man die Funktionswerte an den Grenzen des Wertebereichs, so erhält man:

$$A_i(0) = \frac{65535}{n - K_i(0)} \left(K_i(0) - K_i(0)\right) = \frac{65535}{n - K_i(0)} * 0 = 0 \tag{7}$$

und

$$A_i(65535) = \frac{65535}{n - K_i(0)} \left(K_i(65535) - K_i(0)\right) = 65535 * \frac{n - K_i(0)}{n - K_i(0)} = 65535. \tag{8}$$

Da das kumulative Histogramm $K(j)$ eine monoton wachsende Funktion ist, gilt für alle Werte $0 < j < 65535$:

$$0 \leq A_i(j) \leq 65535. \tag{9}$$

q.e.d

Bestimmung des neuen Grauwerts Im zweiten Durchlauf durch das Bild werden nun die transformierten Grauwerte ins Ausgabebild pro Pixel geschrieben. Ohne Interpolation zwischen den Patches ergibt sich für ein Pixel (x, y), der im Patch i liegt, der Grauwert

$$B_2(x, y) = A_i(B_1(x, y)), \tag{10}$$

wobei B_1 das Eingabe- und B_2 das Ausgabebild sind. Das Ausgabebild zeigt deutlich die Grenzen der Patches, weshalb zwischen den Patches interpoliert werden sollte. Eine Interpolation ist nicht möglich, wenn der Pixel (x, y) in der Nähe einer Bildecke liegt. Am Rand kann die Interpolation nur linear entlang des Randes erfolgen, d.h. zu (x_1, y_1) gibt es einen Patch p_1, dessen Mittelpunkt (x, y) links des Pixels liegt, und einen anderen benachbarten Patch p_2, dessen Mittelpunkt (x_2, y_2) rechts des Pixels liegt. Der interpolierte Grauwert für einen Pixel am oberen oder unteren Bildrand wird wie folgt berechnet:

$$B_2(x, y) = \frac{(x_2 - x) * A_1(B_1(x, y)) + (x - x_1) * A_2(B_1(x, y))}{x_2 - x_1}, \tag{11}$$

analog erfolgt die Interpolation am linken bzw. am rechten Bildrand. Da diese Interpolation ein Spezialfall der bilinearen Interpolation darstellt, wird an dieser Stelle auf den Beweis verzichtet, dass die Resultate stets im Intervall [0..65535] liegen.

In der Bildmitte finden sich zu jedem Pixel (x, y) vier Patches, deren Mittelpunkte ein minimales Rechteck aufspannen, in dem der Pixel liegt. Die Punkte (x_1, y_1) und (x_2, y_2) sind die linke obere bzw. die rechte untere Ecke dieser Bounding Box. Wenn die vier Patches durch die Indices $i = 1, 2, 3, 4$ zeilenweise von oben links nach unten rechts angeordnet werden, dann werden die transformierten Grauwerte mit den Flächenanteilen der anderen Ecke gewichtet.

$$v_1 = (x_2 - x)(y_2 - y) * A_1(B_1(x, y)), \tag{12}$$

$$v_2 = (x - x_1)(y_2 - y) * A_2(B_1(x, y)), \tag{13}$$

$$v_3 = (x_2 - x)(y - y_1) * A_3(B_1(x, y)), \tag{14}$$

$$v_4 = (x - x_1)(y - y_1) * A_4(B_1(x, y)). \tag{15}$$

Für die bilineare Interpolation ergibt sich dann folgende Formel:

$$B_2(x,y) = \frac{v_1 + v_2 + v_3 + v_4}{(x_2 - x_1)(y_2 - y_1)}. \tag{16}$$

Behauptung: Das Ergebnis der bilinearen Interpolation aus Gleichung (**??**) liegt stets im Intervall [0..65535], d.h. die bilineare Interpolation führt zu keinem Under- bzw. Overflow von Grauwerten.

Beweis: Wir betrachten ein Pixel (x, y) in der Bildmitte, so dass die vier Patches in der Umgebung des Pixels existieren. Die Mittelpunkte dieser Patches haben die Koordinaten (x_1, y_1) (oben links), (x_2, y_1) (oben rechts), (x_1, y_2) (unten links) sowie (x_2, y_2) (unten rechts). Diese 4 Patchmittelpunkte spannen ein Rechteck auf, indem das Pixel (x, y) liegt, es gilt also

$$x_1 \leq x \leq x_2 \text{und} y_1 \leq y \leq y_2. \tag{17}$$

$B_1(x, y)$ sei der Grauwert des Pixels im Originalbild, dann ist $g_i = A_i(B_1(x, y))$ der transformierte Grauwert des i-ten Patches, $i = 1 \ldots 4$ zeilenweise von oben links nach unten rechts angeordnet. Diese Grauwerte g_i sind Ergebnisse der Histogrammverebnung, d.h. sie liegen alle im Intervall [0..65535]. Der Grauwert des Pixels (x, y) im Ausgabebild $B_2(x, y)$ wird aus diesen vier Grauwerten g_i als gewichtete Summe berechnet:

$$B_2(x,y) = \sum_{i=1}^{4} \omega_i * g_i. \tag{18}$$

Die Summe der vier Gewichte ω_i ist 1, weil gilt:

$$\omega_1 + \omega_2 + \omega_3 + \omega_4 = \frac{\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4}{(x_2 - x_1)(y_2 - y_1)} \tag{19}$$

und

$$\begin{aligned} \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 &= (x_2 - x)(y_2 - y) + (x - x_1)(y_2 - y) + \\ &\quad (x_2 - x)(y - y_1) + (x - x_1)(y - y_1) \\ &= x_2y_2 - x_2y - xy_2 + xy + xy_2 - xy - x_1y_2 + x_1y + \\ &\quad x_2y - x_2y_1 - xy + xy_1 + xy - x_1y_2 - x_1y + x_1y_1 \\ &= x_2y_2 - x_1y_2 - x_2y_1 + x_1y_1 \\ &= (x_2 - x_1)(y_2 - y_1). \end{aligned} \tag{20}$$

Wegen $g_i \leq 65535$ können wir die transformierten Grauwerte g_i nach oben hin abschätzen, es gilt also:

$$B_2(x,y) = \sum_{i=1}^{4} \omega_i * g_i \leq \sum_{i=1}^{4} \omega_i * 65535 = 65535 * \sum_{i=1}^{4} \omega_i = 65535. \tag{21}$$

Die untere Schranke ist trivialerweise 0, da aus $g_i \geq 0$ folgt:

$$B_2(x,y) = \sum_{i=1}^{4} \omega_i * g_i \geq \sum_{i=1}^{4} \omega_i * 0 = \sum_{i=1}^{4} 0 = 0. \tag{22}$$

Da die Summe eine lineare Abbildung ist, folgt aus den beiden Schranken: Der Grauwert des Pixels (x, y) im Ausgabebild liegt im Intervall [0..65535].

q.e.d

Stauchung des verebneten Histogramms Die verebnung des Histogramms hat einen starken Kontrast im neuen Bild zur Folge. Dieser Kontrast kann durch eine zusätzliche Stauchung des verebneten Histogramms abgeschwächt werden. Wenn sich die Grauwerte im verebneten Histogramm über das Intervall [0..65535] erstrecken, dann wird die Spannweite der belegten Grauwerte nach der Stauchung um einen Faktor k kleiner sein.

Dieser Faktor k wird aus den Gradienten ∇B_2 im Bild mit dem verebneten Histogramm hergeleitet. Unter konstanter Fortführung des Bildes am Bildrand kann für jedes Pixel (x, y) der Gradient durch Anwendung der Sobel-Filter

$$S_r = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \text{und} \quad S_c = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{23}$$

wie folgt berechnet werden:

$$\nabla B_2 = \sqrt{(S_r * B_2)^2 + (S_c * B_2)^2}. \tag{24}$$

Für jedes Bildpatch werden nun der maximale Gradient m_1 sowie der Median m_2 der Gradienten berechnet. Mit diesen beiden Werten kann nun die lineare Transformation, die Stauchung, des Histogramms bestimmt werden. Wegen $0 \leq m_{2i} \leq m_{1i}$ ergibt sich für den Stauchungsfaktor im Patch i

$$k_i = \frac{m_{1i} - m_{2i}}{m_{1i}} \tag{25}$$

ein Wert aus dem Intervall [0..1]. Somit ist sichergestellt, dass sich die Spannweite der im Bild belegten Grauwerte nicht vergrößern kann, sondern i. d. R. nur verringert. Um die Stauchung des Histogramms auf ein sehr kleines Intervall oder einen einzigen Grauwert zu verhindern, wird jedes k_i auf eine Mindestgröße hin überprüft, d. h. es gilt immer $k_i > s > 0$, wobei s ein vom Benutzer festgelegter Schwellwert ist.

Das neue Bild B_3 ergibt sich nun durch Anwendung folgender Punkttransformation auf dem verebneten Bild B_2:

$$B_3(x, y) = k * (B_2(x, y) - 32768) + 32768. \tag{26}$$

Durch diese Abbildung bleibt der Mittelwert des Bildes erhalten, die Streuung der Grauwerte verringert sich um k. Werden Grauwerte bei dieser Abbildung auf einen Wert außerhalb des 16-Bit-Bereichs transformiert, so werden diese auf 0 bzw. 65535 gesetzt, d. h. es tritt ein Informationsverlust auf. Der Prozentsatz dieser betroffenen Grauwerte wird im Ausgabemodus des Programms angezeigt. Damit die Patchgrenzen im Ausgabebild nicht sichtbar werden, wird (sofern vom Benutzer gewünscht) die Histogrammstauchung zwischen den Patches analog zur Verebnung bilinear interpoliert.

Programmende Vor Abschluss des Programms wird das Ausgabebild gespeichert. Bei Verringerung der radiometrischen Auflösung auf 8 Bit muss ein neues Bild angelegt werden, die Umrechnung erfolgt mit dem `shift`-Operator, d. h. die ersten 8 Bits der 16-Bit-Grauwerte bestimmen den Grauwert des Ausgabebilds.

4 Beispiele

4.1 1. Beispiel mit Histogrammen

4.2 2. Beispiel ohne Histogrammen

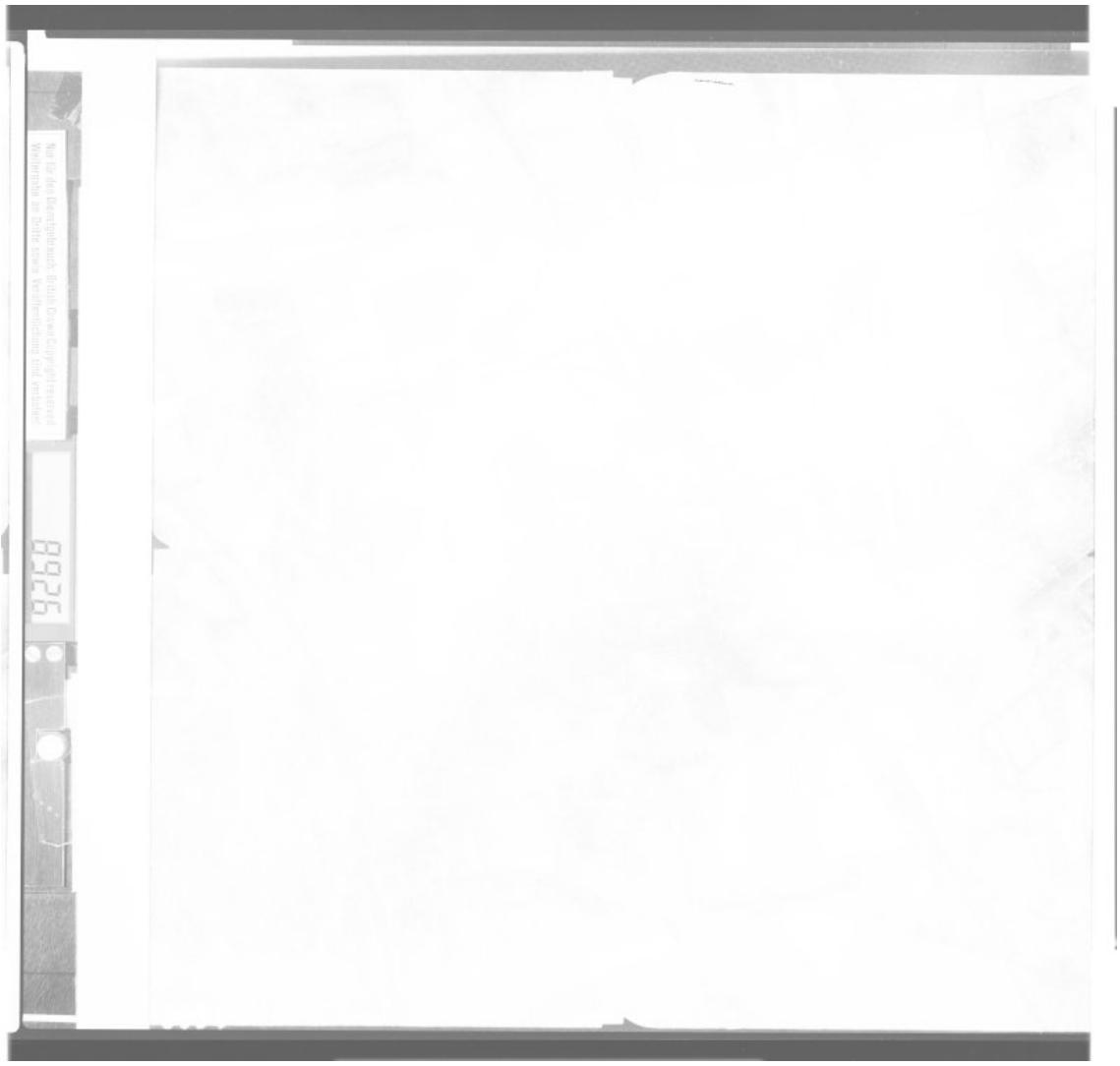

Figure 1: Eingangsbild (Darstellung in 8 Bits pro Pixel statt mit 16 Bits pro Pixel)

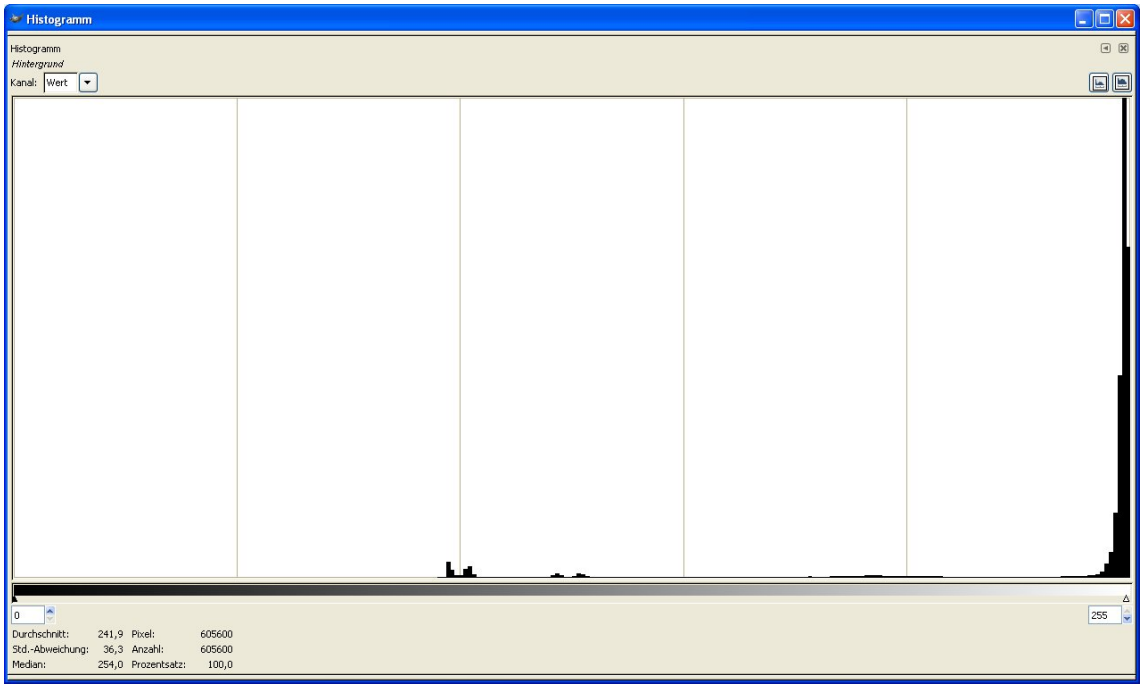

Figure 2: Histogramm des Eingangsbilds (nach Konvertierung in 8 Bits pro Pixel)

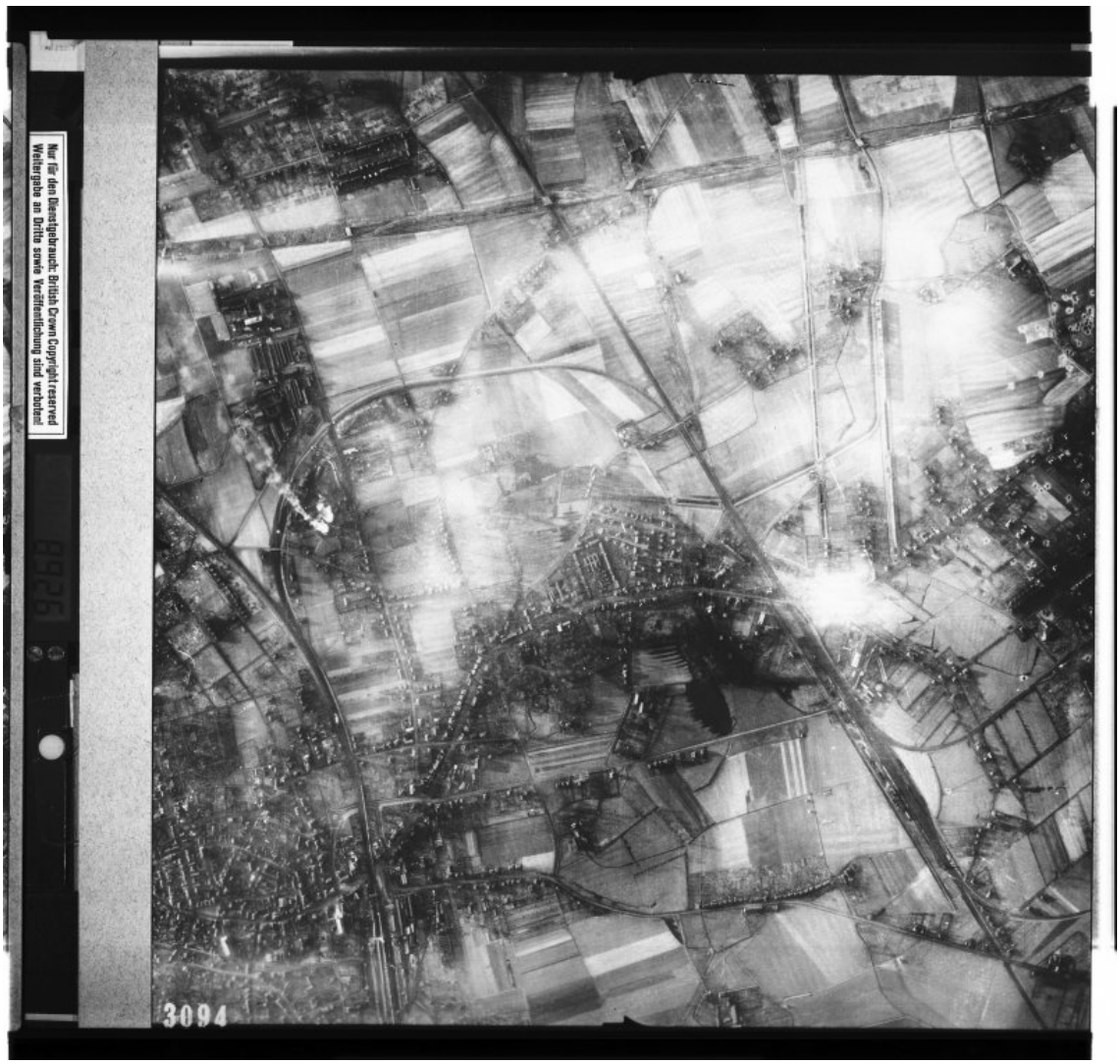

Figure 3: Bildverbesserung mit Globalem Dodging

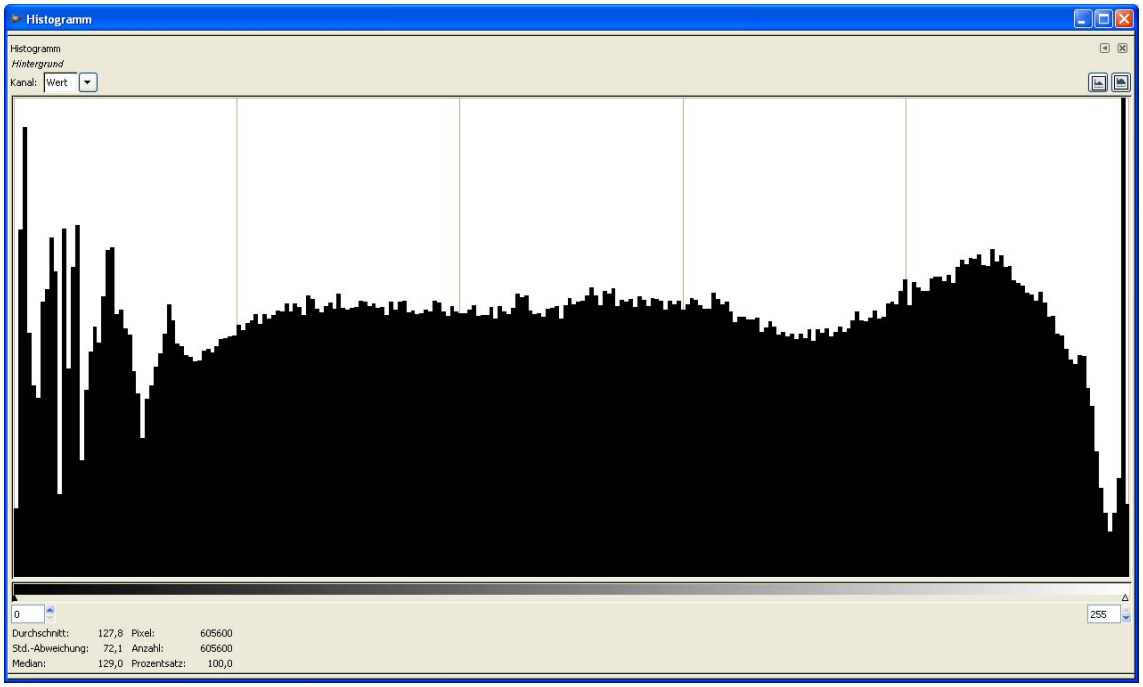

Figure 4: Histogramm des global gedodgten Bilds

Figure 5: Bildverbesserung mit zeilenweisem Dodging ohne Interpolation

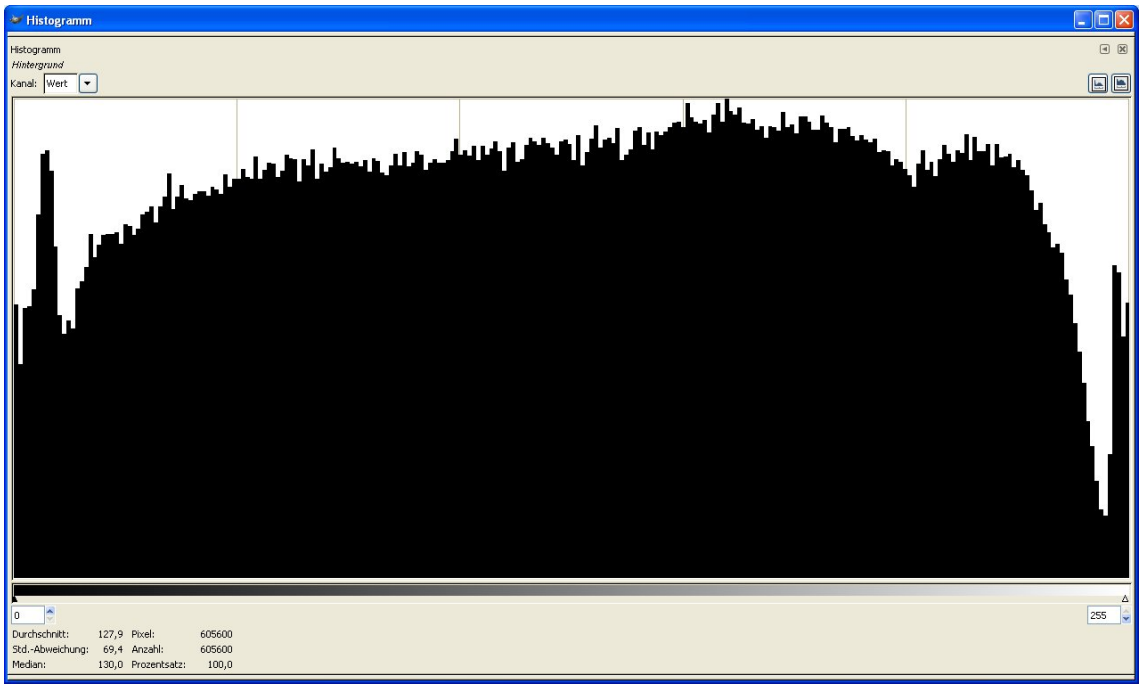

Figure 6: Histogramm des zeilenweise gedodgten Bilds (ohne Interpolation)

Figure 7: Bildverbesserung mit zeilenweisem Dodging mit Interpolation

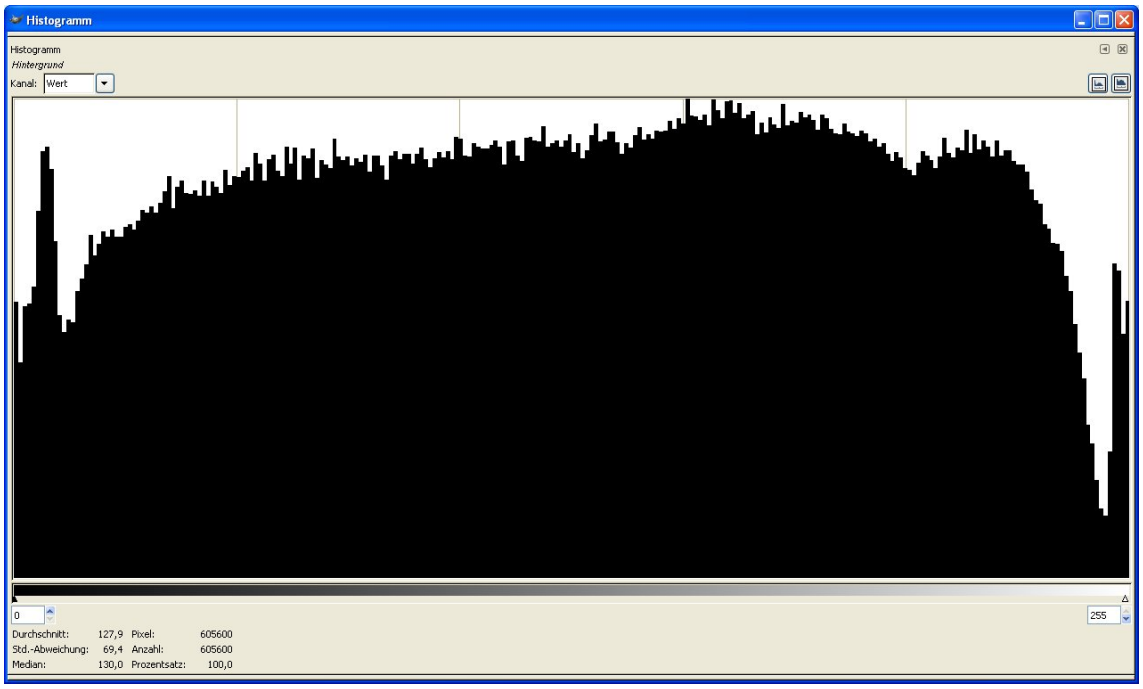

Figure 8: Histogramm des zeilenweise gedodgten Bilds (mit Interpolation)

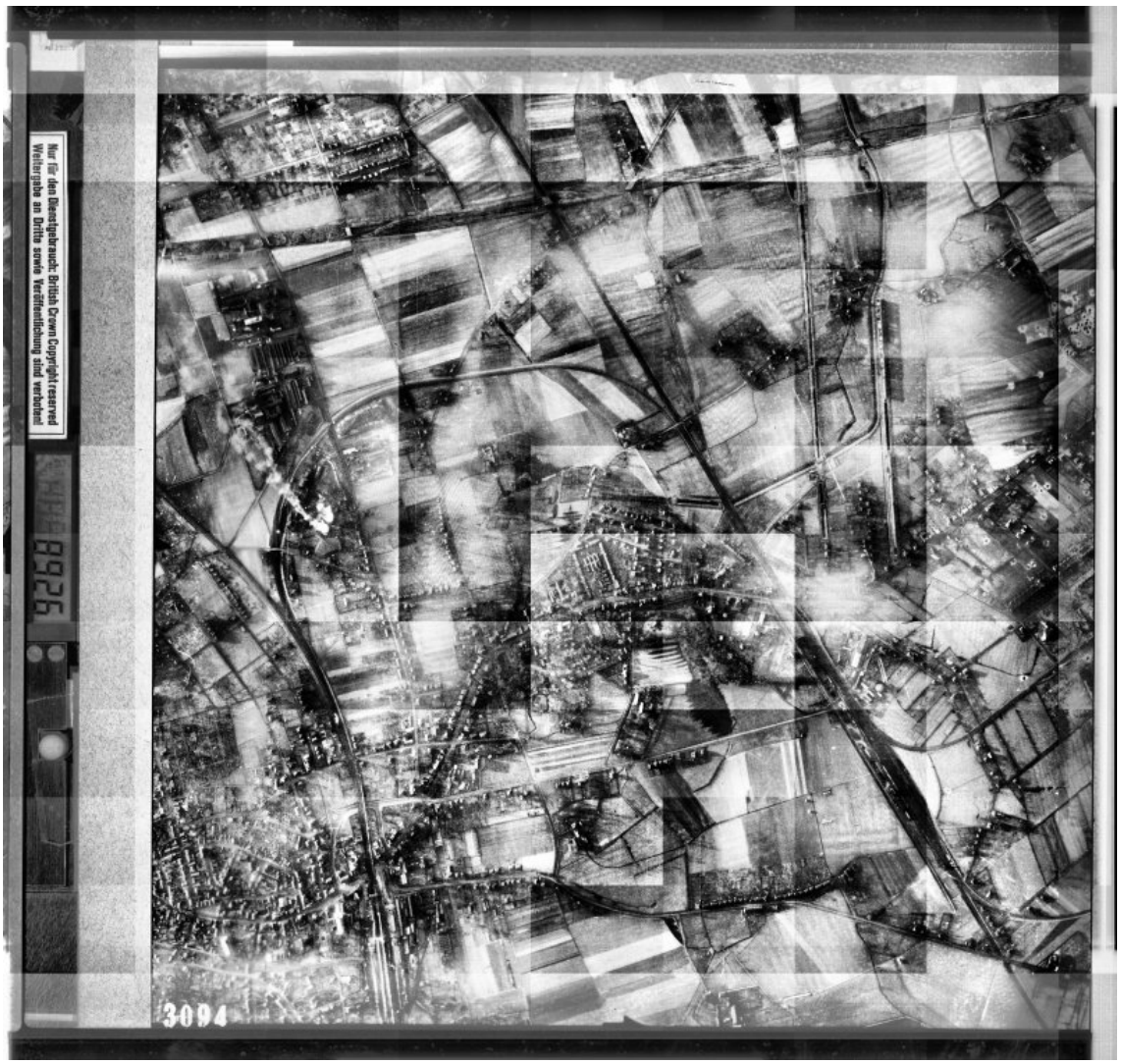

Figure 9: Bildverbesserung mit patchweisem Dodging ohne Interpolation

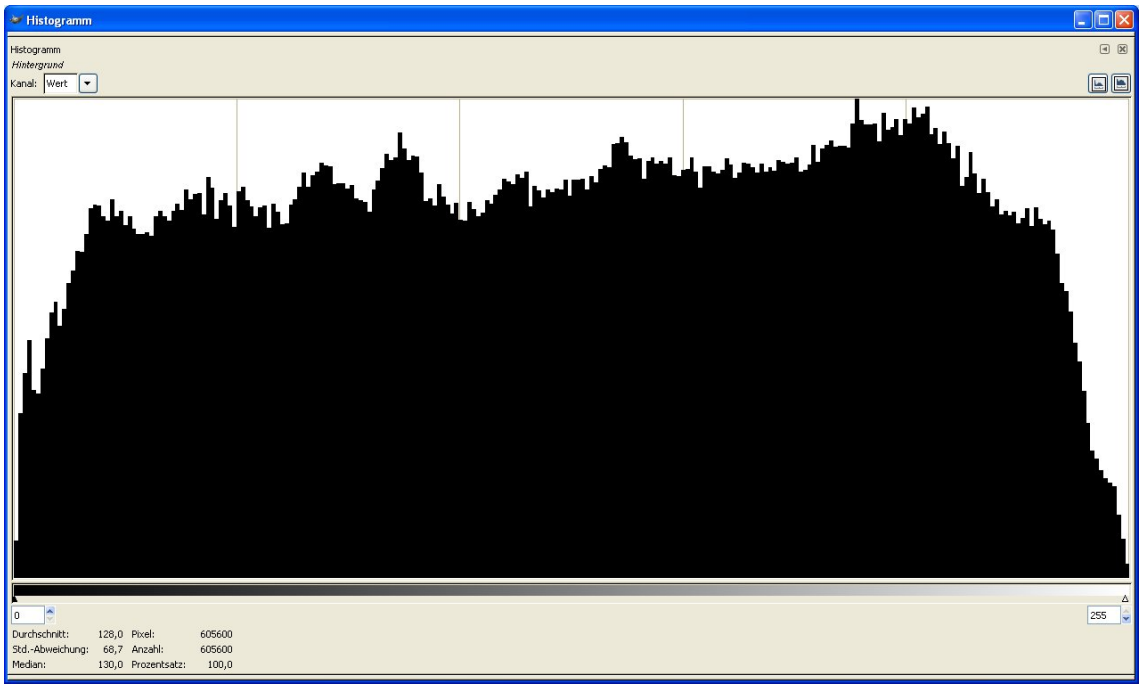

Figure 10: Histogramm des patchweise gedodgten Bilds (ohne Interpolation)

Figure 11: Bildverbesserung mit patchweisem Dodging mit Interpolation

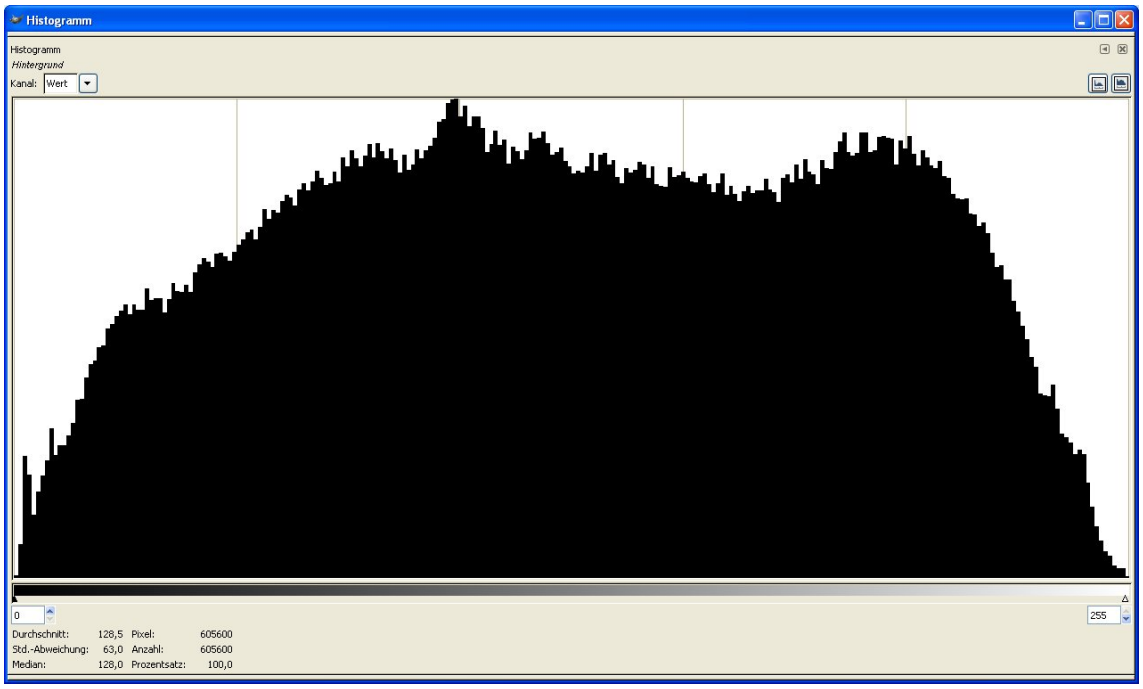

Figure 12: Histogramm des patchweise gedodgten Bilds (mit Interpolation)

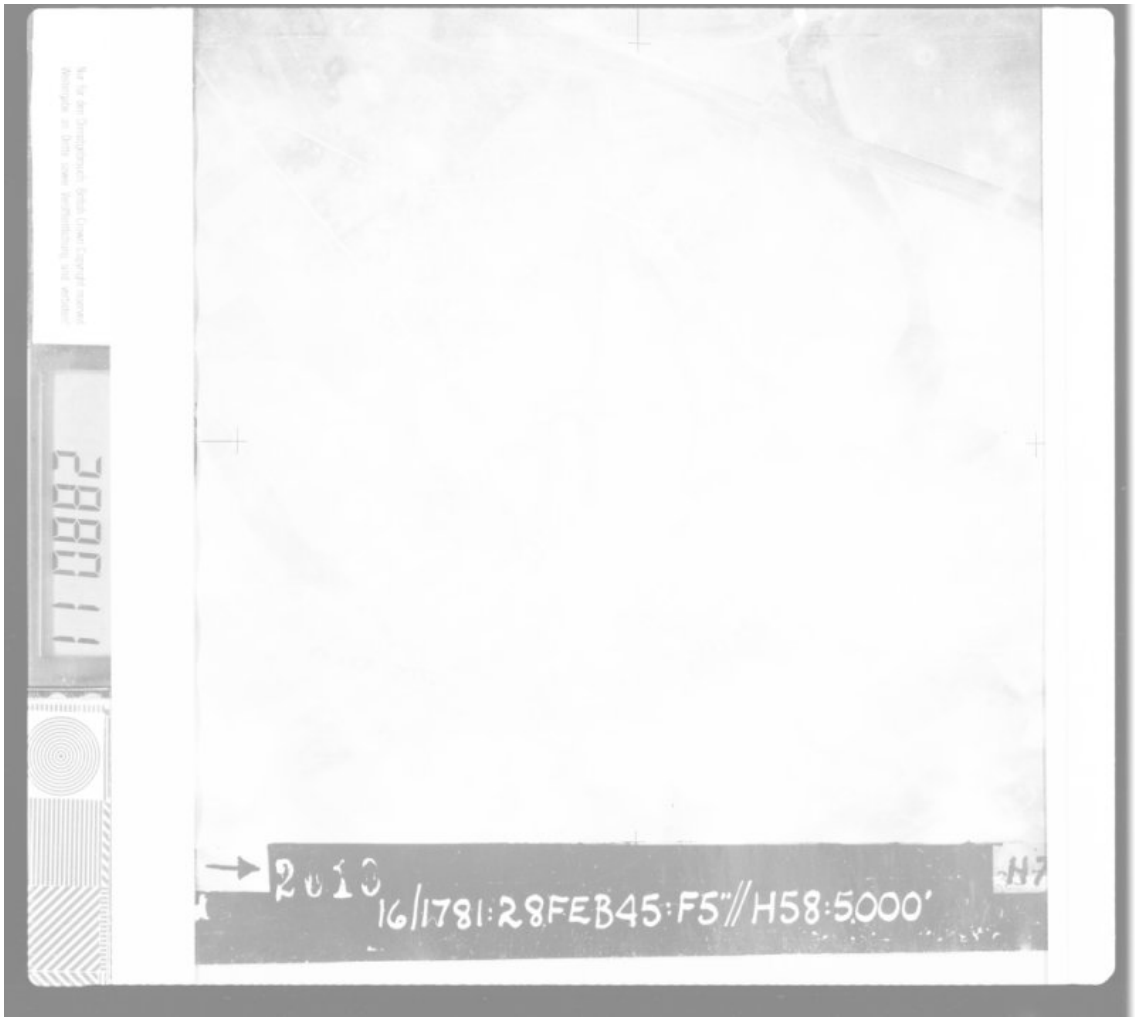

Figure 13: Eingangsbild (Darstellung in 8 Bits pro Pixel statt mit 16 Bits pro Pixel)

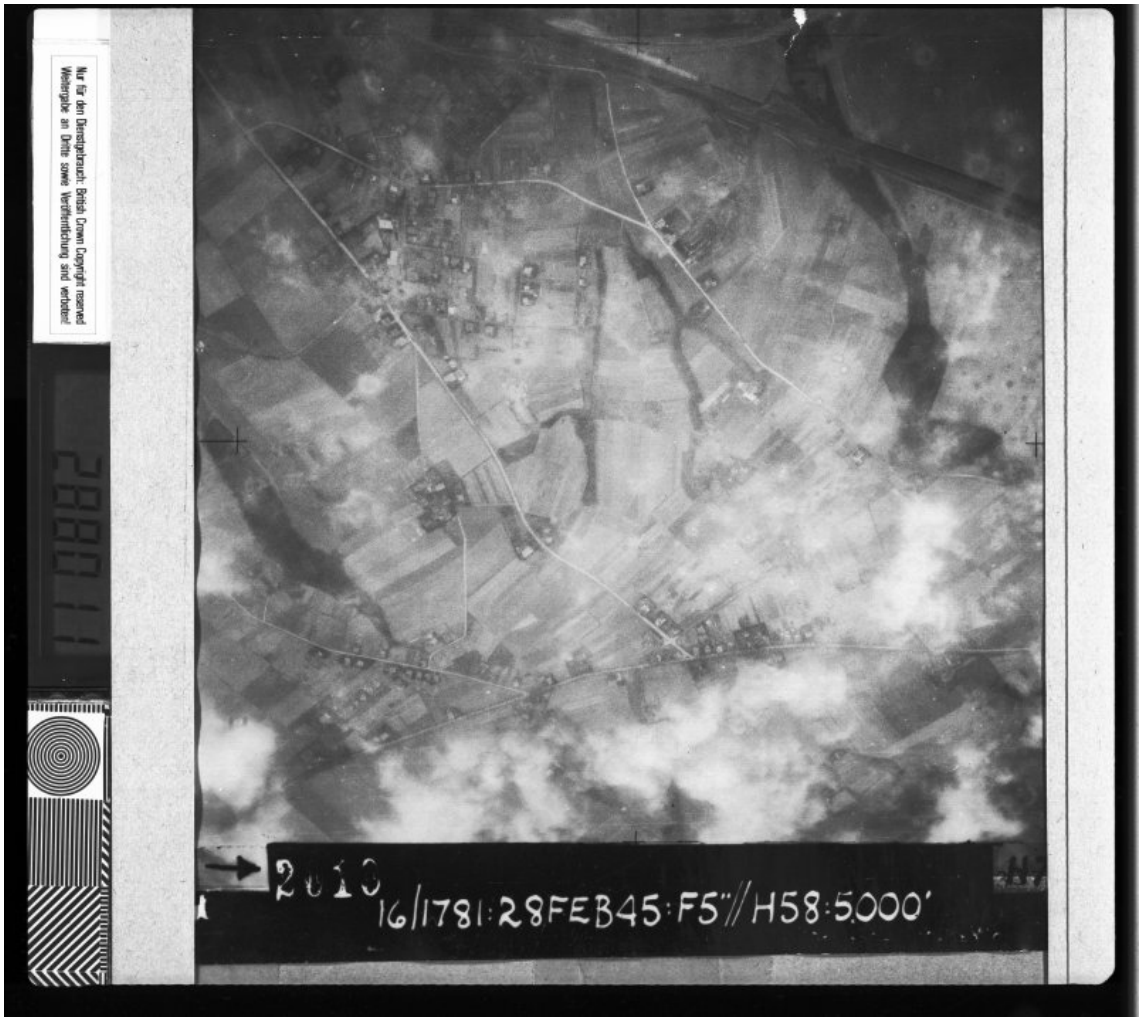

Figure 14: Bildverbesserung mit Globalem Dodging

Figure 15: Bildverbesserung mit zeilenweisem Dodging ohne Interpolation

Figure 16: Bildverbesserung mit zeilenweisem Dodging mit Interpolation

Figure 17: Bildverbesserung mit patchweisem Dodging ohne Interpolation

Figure 18: Bildverbesserung mit patchweisem Dodging mit Interpolation