

Bildverarbeitung mit IDL und ENVI[®]

Spezialseminar „Digitale Bildverarbeitung und Mustererkennung in der Fernerkundung“

Leiter: Dr. Mort CANTY
Referent: Carsten KOLENDA

kolenda@uni-bonn.de

Grundlagen

Fernseh-Kameras ... (Landsat)

- RBV, *Return Beam Vidicon*: im Prinzip eine Fernseh-Kamera
- Auf Landsat 1, 2, 3 eingesetzt
- Erfassung des Gesamtbildes auf photosensitivem Schirm

... opto-mechanische Abtaster ... (Landsat)

- MSS, *Multi Spectral Scanner*: Landsat 1-5
- TM, *Thematic Mapper*: Landsat 4-5
- ETM+, *Enhanced Thematic Mapper plus*: Landsat 7

... und opto-elektronische Abtaster (SPOT)

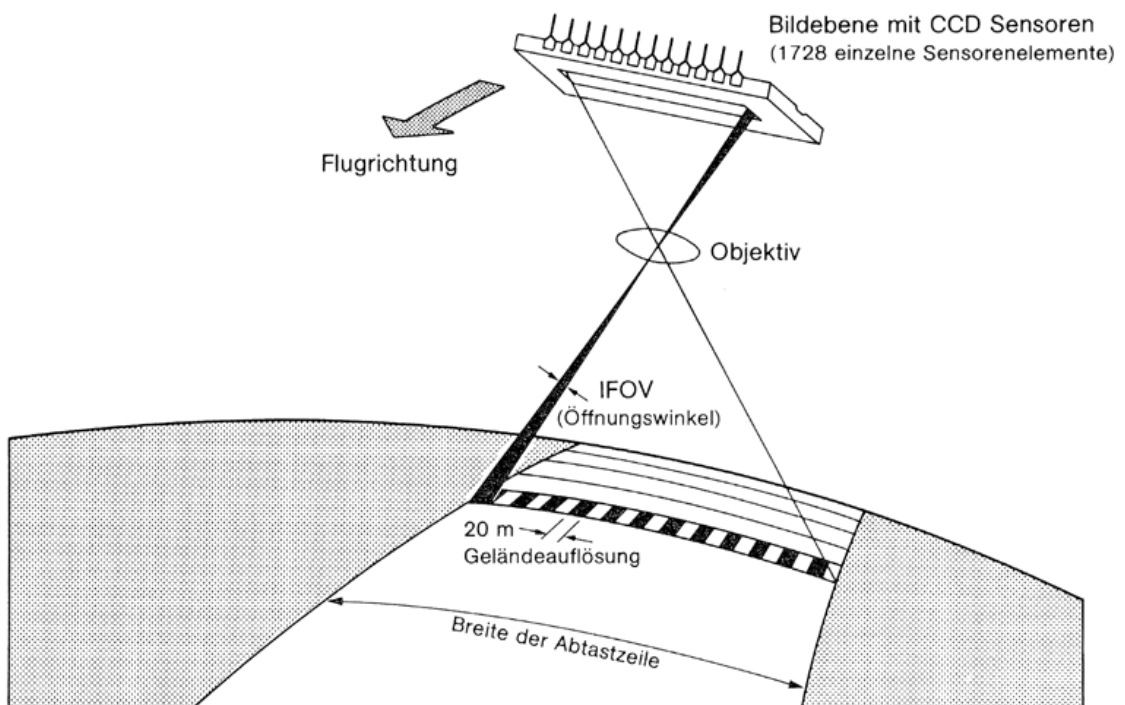


Abbildung 1: CCD-Sensor im SPOT-Satellitensystem. Quelle: LÖFFLER 1994:48

Bildelemente (*Pixels*) in einem Datenfeld (*Array*)

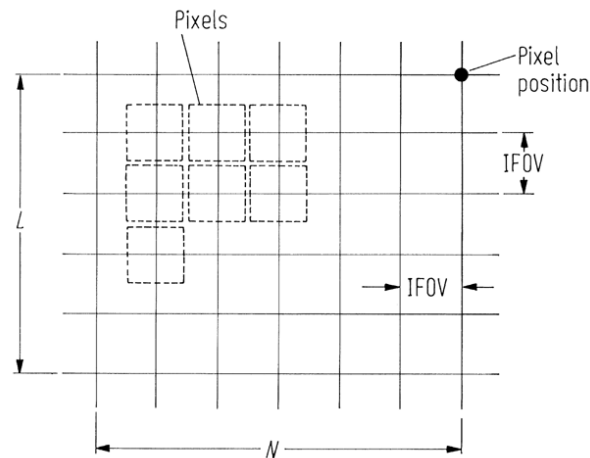


Abbildung 2: Pixelgröße, -anordnung und Instantaneous Field of View (IFOV). Quelle: RICHARDS/JIA 1999:49

Jeweils ein Datenfeld für jeden Kanal

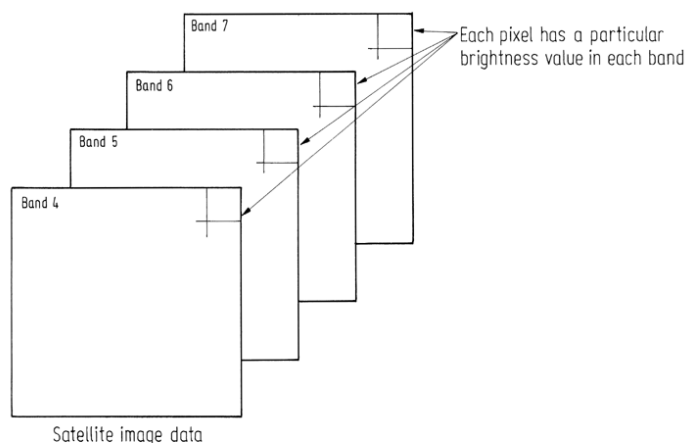


Abbildung 3: Speicherung der Bilddaten getrennt nach Spektralbändern. Quelle: RICHARDS/JIA 1999:84

Verarbeitung der Bilddaten auf dem Rechner

- Die Bilder liegen im Rechner als eines oder mehrere **Datenfelder** vor.
- Diese Daten müssen zunächst radiometrisch und geometrisch **korrigiert** werden.
- Danach können die Daten **weiterverwendet** werden, etwa zur Klassifikation.
- Alle diese Operationen sind letztlich **Manipulationen von Datenfeldern**.

IDL

Schwerpunkte von IDL

- **IDL: Interactive Data Language**, Datenfeld-orientierte Programmier-sprache für wissenschaftlichen Einsatz
- Enthält unter anderem Werkzeuge für mathematische Berechnungen und graphische Darstellung

Einfache Aufgaben mit Bildern (in IDL)

- Ein Satellitenbild **öffnen und darstellen**
- **Störungen entfernen** mit einem Median-Filter
- **Kontrast verbessern** mit *Histogram Equalization*

Dem Rechner einen Befehl geben ...

- Die Bedienung eines Rechners durch graphische Oberflächen (Menüs, Schaltflächen, Symbole...) ist bekannt
- Aber auch: Kommandozeile, durch die Befehle in Textform gegeben werden (z. B. Windows Eingabeaufforderung)
- Beispiel:
Rename Vortrag.DOC Referat.DOC
- Auch IDL hat eine Kommandozeile und erhält vom Nutzer Befehle in Textform
- Beispiel:
window, 1, xSize=500, ySize=300
Öffnet ein Fenster (etwa zur Bilddarstellung) mit dem Index 1, 200 Pixel Breite und 300 Pixel Höhe

Aufgabe #1: Bild öffnen und darstellen

- Die Datei hanford1983r_band4.bmp enthält — zur vereinfachten Darstellung der nachfolgenden Filter-Beispiele — nur Band 4 (nahes Infrarot) aus dem Satellitenbild.
- Wir werden jetzt zunächst IDL und dann das Bild öffnen. Anschließend stellen wir das Bild auf dem Schirm dar.

```
filename = Filepath('hanford1983r_band4.bmp', SubDir=['seminar'])
ok = Query_BMP(filename, fileinfo)
image = Read_BMP(filename)
window, 0, xSize=750, ySize=850
TV, image
```

- Mit einer Reihe von Befehlen in der IDL-Kommandozeile haben wir den Speicherort der **Bilddatei beschrieben**, die Datei **geprüft** und **geladen**. Dann haben wir ein **Fenster geöffnet** und die geladenen Daten im Fenster **dargestellt**.

Aufgabe #2: Störungen entfernen (Median-Filter)

- Wir werden jetzt **künstlich Störungen in das Bild einfügen** (dabei sehen wir auch, wie bequem IDL den Zugriff auf Datenfelder erlaubt).
- Anschließend verwenden wir einen **Median-Filter**, um die Störungen wieder zu entfernen.
- Eine Schablone bestimmter Größe (wie hier 3 x 3 Pixel) wird Pixel für Pixel über das Bild geschoben
- Der Pixel in der Mitte bekommt den Median-Wert aller von der Schablone abgedeckten Pixel
- V. a. für impulsartige Störungen geeignet

```
noisy = image
points = RandomU(seed, 20000) * 750 * 850
noisy(points) = 255
points = RandomU(seed, 20000) * 750 * 850
noisy(points) = 0
TV, noisy
TV, Median(noisy, 3)
TV, Median(noisy, 2)
TV, Median(noisy, 10)
TV, noisy
image2 = Median(noisy, 3)
TV, image2
```

- Mit einer Reihe von Befehlen in der IDL-Kommandozeile haben wir **Ausreißer-Werte in das Bild eingefügt**. Dann haben wir verschiedene Einstellungen für den **Median-Filter probiert** und mit der geeignetsten Wahl die Fehler aus dem Bild wieder entfernt.
- Siehe auch: RICHARDS/JIA 1999:121–122

Aufgabe #3: Kontrast verbessern (*Histogram Equalization*)

- Als letzte Aufgabe werden wir eine häufig durchgeführte Operation ausführen: die Verbesserung des Bildkontrasts. Der hier gewählte Ansatz ist die *Histogram Equalization*.
- Die Verteilung der Pixelwerte im Bild ist oftmals nicht gleichmäßig, sondern gehäuft
- Wären alle Pixelwerte gleich-häufig (Histogramm mit gleich hohen Balken), dann wäre der Kontrast für jeden Helligkeitsbereich des Bildes optimal

```
TV, image2
Plot, Histogram(image2)
TV, Hist_Equal(image2)
Plot, Histogram(Hist_Equal(image2))
TV, Hist_Equal(image)
```

- Mit einem Befehl haben wir eine **Histogram Equalization** mit dem Bild ausgeführt. Dadurch konnten wir den Kontrast stark erhöhen.
- Siehe auch: RICHARDS/JIA 1999:97–102

Mehrere Befehle hintereinander geben ...

- Um eine Aufgabe zu lösen, die sich aus Einzelschritten zusammensetzt, haben wir **mehrere Befehle** hintereinander gegeben
- Beispiel (Zeichnet einen Zufalls-Graph):

```
window, 1, XSize=500, YSize=300
Zufall = RandomU(Seed, 25)
Plot, Zufall
```
- Wir können IDL automatisch die Befehlsfolgen einer Arbeitssitzung in eine Textdatei — hier Journal-Datei genannt — **protokollieren** lassen.
- Dazu geben wir das Kommando:

```
Journal, 'Dateiname'
```
- IDL kann alle Befehle einer Journal-Datei im sogenannten **Stapelverarbeitungs-betrieb** zu einem späteren Zeitpunkt erneut ausführen.
- Dazu geben wir das Kommando:

```
@Dateiname
```

 (also den Namen der Journal-Datei mit vorangestelltem @-Symbol)

... viele Befehle zusammen ergeben ein Programm

- Ein **Programm** ist ähnlich einer Stapelverarbeitungsdatei — also prinzipiell nicht viel mehr als eine Auflistung mehrerer Befehle. Es endet jedoch immer mit dem Statement END.
- **IDLDE**: Entwicklungsumgebung für IDL (*IDL Development Environment*)

Programmiergrundlagen

- **1: Variablen**
Information (zwischen-) speichern
- **2: Kontrollstrukturen**
Programmablauf steuern
- **3: Prozeduren und Funktionen**
Programm strukturieren und modularisieren

Programmiergrundlagen 1: Variablen

- Beim Durchführen von Berechnungen müssen häufig **Werte vorübergehend gespeichert** werden
- **Variablenamen** müssen mit einem **Buchstaben** beginnen und dürfen darüber hinaus Buchstaben, Ziffern, Unterstriche (_) und Dollar-Zeichen (\$) enthalten
- Attribute einer Variable
 - **Datentyp**
 - **Organisationsstruktur**
- Es gibt **14 einfache Datentypen** in IDL. Ein Datentyp beschreibt, welche Art von Daten gespeichert werden können
- Folgende Organisationsstrukturen sind in IDL für Variablen möglich:
 - **Skalar** (*scalar*)
 - Einzelner Wert


```
test_123 = 573L
testB = 123.45
```
 - **Vektor** (*vector*)
 - Matrix mit nur einer Zeile bzw. Spalte


```
versuch33 = [ 1, 3 ]
test02 = [ [1], [4], [8] ]
beispiel_C = FIndGen (2)
```
 - **Feld** (*array*)
 - Matrix mit mehr als einer Zeile bzw. Spalte


```
test98 = [ [7, 5], [4, 6] ]
probe123 = ULIndGen (5, 7)
```
 - **Struktur** (*structure*)
 - Zusammengesetzte Organisationsstruktur


```
struct = { student, $
          matrikelnr: 0L, alter: 0 }

...
carsten = { student }
carsten.matrikelnr = 1309516L
carsten.alter = 24
```

Programmiergrundlagen 2: Kontrollstrukturen

- Kontrollstrukturen ermöglichen das Steuern des Programmablaufs
- Ohne Anweisungen zur Ablaufsteuerung durchläuft die Logik eines Programms Anweisungen „von oben nach unten“ (also sequentiell)
- Mächtigkeit und Nutzen eines Programms sind eng mit der Fähigkeit verbunden, die Reihenfolge von Anweisungen mit
 - **Entscheidungsstrukturen** und
 - **Schleifenstrukturen**
 zu ändern
- **Entscheidungsstrukturen**
 - **Wenn** (Bedingung) **dann** (Möglichkeit 1) **sonst** (Alternative)


```
IF ... THEN ... ELSE
```
 - **Mehrfach-Fallunterscheidung**
 - CASE
 - SWITCH

■ IF ... THEN ... ELSE

Wird verwendet, um eine oder mehrere Anweisungen bedingt auszuführen

```
IF (num GT 10) THEN BEGIN
    index = 2
    num = 0
ENDIF ELSE BEGIN
    index = 4
    num = -10
ENDELSE
```

■ CASE

Wird verwendet, um aus mehreren Anweisungsblöcken einen Block selektiv auszuführen

```
CASE num OF
    0: x = 47
    1: x = 11
    2: x = 14
    ELSE: x = 18
ENDCASE
```

■ SWITCH

Wird verwendet, um aus mehreren Anweisungsblöcken einen oder mehrere Blöcke selektiv auszuführen

```
SWITCH var OF
    test * 3: x = 47
    test + 8: x = 11
    test - 5: x = 14
    ELSE: x = 18
ENDCASE
```

■ Schleifenstrukturen**– Schleife mit Zählvariable**

FOR

– Schleifen ohne Zählvariable

■ WHILE

■ REPEAT ... UNTIL

■ FOR

Wird verwendet, um eine oder mehrere Anweisungen mehrmals auszuführen — und zwar, wenn vorhergesagt werden kann, wie oft

```
a = 1
FOR j=1, 10, 2 DO BEGIN
    Print, j
    a = a * j * 2
ENDFOR
```

■ WHILE

Wird verwendet, um eine oder mehrere Anweisungen mehrmals auszuführen — und zwar, solange eine Bedingung erfüllt bleibt

```
WHILE (number LT 1000) DO BEGIN
    step = step + 3
    number = number + step
ENDWHILE
```

■ REPEAT ... UNTIL

Wird verwendet, um eine oder mehrere Anweisungen mehrmals auszuführen — und zwar, bis eine Bedingung erfüllt wird

```
REPEAT BEGIN
    step = step + 3
    number = number + step
ENDREP UNTIL (number GE 1000)
```

Programmiergrundlagen 3: Prozeduren und Funktionen

- Prozeduren und Funktionen helfen, ein Programm in kleinere logische Komponenten zu unterteilen.
- Die Komponenten können als Bausteine zur Erweiterung von IDL verwendet werden — sie werden dann auf dieselbe Art und Weise eingesetzt, wie die bereits mit IDL gelieferten Befehle

■ Prozeduren

- Prozeduren haben keinen Rückgabewert
TV, image123

■ Funktionen

- Funktionen haben einen Rückgabewert
y = sin (0.5)

- Beispiel einer **Prozedur mit Parameter**
PRO Greetings, name
 Print, "Hallo, " + name
END

```
Greetings, "Heini Müller"  
END
```

- Beispiel einer **Funktion**
FUNCTION Doubleval, value
 RETURN, value * 2
END

```
Print, Doubleval (5.3)  
END
```

- Der **Gültigkeitsbereich** (*scope*) einer Variable ist auf diejenige Prozedur bzw. Funktion beschränkt, in der die Variable erzeugt wird.
- Bei Beendigung der Prozedur bzw. Funktion wird der Speicherplatz für diese Variable wieder freigegeben.

ENVI

Schwerpunkte von ENVI

- **ENVI**: *Environment for Visualizing Images*, Bildverarbeitungssystem für wissenschaftlichen Einsatz
- Insbesondere geeignet für Verarbeitung von Satelliten- und anderen Fernerkundungs-Bildern
- In **IDL** programmiert

Hauptkomponentenanalyse

- Multispektrale Daten-Bänder weisen oft eine **große Korrelation** auf.
- Mit Hilfe der **Hauptkomponentenanalyse (Principal Component Analysis)** kann man ...
 - nicht korrelierte Ausgabe-Bänder erzeugen
 - Rauschen und benötigte Information voneinander trennen
- PC-Bänder sind **lineare Kombinationen** der ursprünglichen Spektral-Bänder
- Nach Ausführung der PC-Transformation liegt **dieselbe Anzahl Bänder** wie zuvor vor
- Die Daten im ersten PC-Band enthalten die **größte Varianz**, die Daten im zweiten PC-Band die Zweitgrößte usf.
- Die **letzten PC-Bänder** sind **verrauscht**, weil sie nur noch eine kleine Varianz aufweisen. Ursache ist das Rauschen in den ursprünglichen Spektraldaten

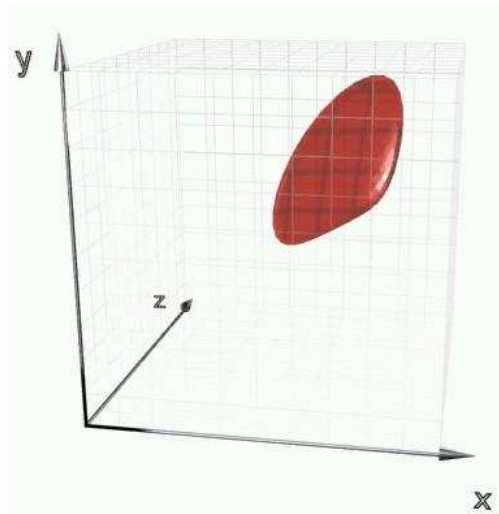


Abbildung 4: Stichprobenelemente im dreidimensionalen Raum vor der PC-Transformation.
Quelle: <http://www.digimusik.de/PCA/bsp3d.html>

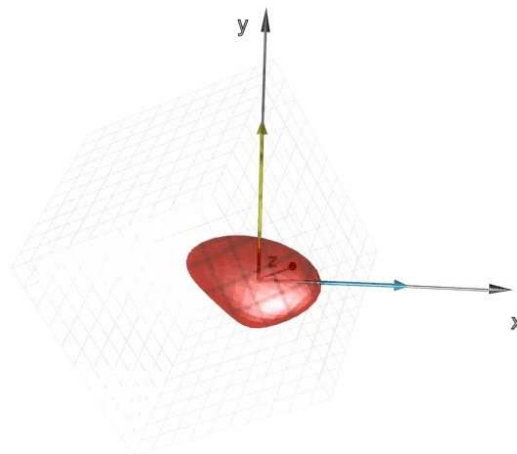


Abbildung 5: X- und Y-Achse sind nach größter und zweitgrößter Varianz ausgerichtet. Der Informationsverlust bei Weglassen der Z-Achse wäre gering. Quelle: <http://www.digimusik.de/PCA/bsp3d.html>

```
File > Open Image File  
(Datei herausuchen)  
Öffnen  
RGB Color  
(Zuordnen: R – Band 3, G – Band 2, B – Band 1)  
Load RGB  
Transform > Principal Components > Forward PC Rotation > Compute...  
OK  
Output Result is > Memory
```

- Nach diesen Schritten kann im Fenster „Available Bands List“ ausgewählt werden, die PC-Bänder darzustellen.

Literatur

- FANNING, David (2000²): *IDL Programming Techniques*. Fort Collins.
- LÖFFLER, Ernst (1994²): *Geographie und Fernerkundung*. Eine Einführung in die geographische Interpretation von Luftbildern und modernen Fernerkundungsdaten. Stuttgart.
- RICHARDS, John u. Xiuping JIA (1999³): *Remote Sensing Digital Image Analysis*. An Introduction. Berlin, Heidelberg, New York.
- <http://landsat7.usgs.gov/>
Homepage des Landsat-7-Programms der United States Geological Survey
- <http://www.rsinc.com/>, <http://www.creaso.com/>
Research Systems Inc., Hersteller von IDL und ENVI
- <http://www.kis.uni-freiburg.de/~dobler/doc/idl/>
Einführung in IDL
- http://nstx.pppl.gov/nstx/Software/IDL/idl_intro.html
Einführung in IDL vom National Spherical Torus Experiment (U.S.)