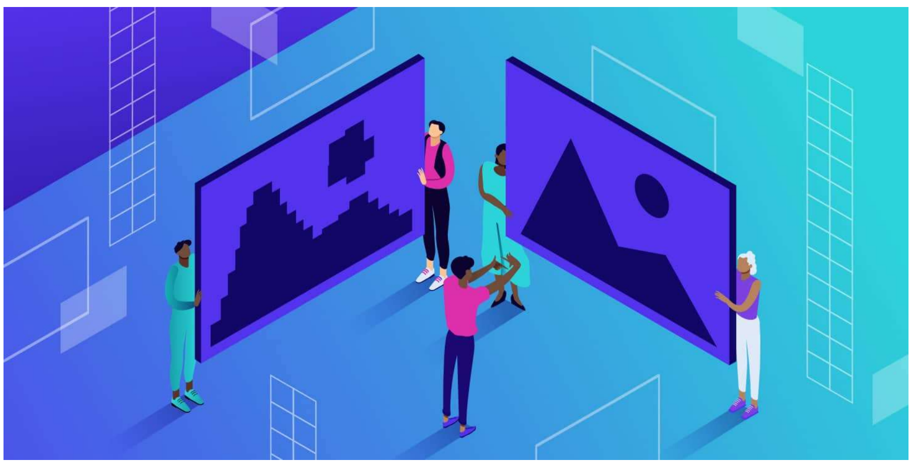
FFHS
Fernfachhochschule
Schweiz
Mitglied der SUPSI



Signal-/Bilddatenkomprimierung

Verfasser:Kaan Kaplan

Dozent: Jörg Osterrieder

Ort und Abgabedatum: Bern, 31. Dezember 2024

Einleitung

Die Datenkomprimierung spielt eine zentrale Rolle in der modernen Informationsverarbeitung, besonders wenn es um Bilder und Signale geht. Ein häufig genutztes Verfahren in der Bildkomprimierung ist das JPEG-Format, das die diskrete Kosinustransformation (DCT) als mathematische Grundlage verwendet. Die Hauptidee hinter der JPEG-Komprimierung ist es, Bilddaten so zu reduzieren, dass redundante oder kaum wahrnehmbare Informationen entfernt werden, ohne die wahrgenommene Bildqualität stark zu beeinträchtigen. Ein gutes Beispiel dafür ist das gezielte Entfernen oder Komprimieren von hochfrequenten Bilddetails, die vom menschlichen Auge kaum wahrgenommen werden.

Die DCT ist hierbei entscheidend, da sie Bilddaten aus ihrer räumlichen Darstellung in den Frequenzbereich transformiert. Dadurch lassen sich Bildinhalte effizienter darstellen und weniger relevante Informationen einfacher herausfiltern. Zum Beispiel werden die niedrigen Frequenzen, die die Hauptinformationen des Bildes ausmachen, stärker priorisiert, während hochfrequente Komponenten, die oft nur Details enthalten, reduziert werden. Diese Eigenschaft macht die DCT zu einem unverzichtbaren Werkzeug für verlustbehaftete Komprimierungsverfahren, die sowohl platzsparend als auch qualitativ überzeugend sind.

Theoretischer Teil

Definition und Eigenschaften

Die diskrete Kosinustransformation (DCT) ist ein mathematisches Verfahren, das in der Signal- und Bildverarbeitung weit verbreitet ist. Sie gehört zur Familie der Fourier-Transformationen und wird verwendet, um Daten von der Zeit- bzw. Raumdomäne in den Frequenzbereich zu überführen. Die DCT ist besonders effektiv, da sie die Energie eines Signals auf eine kleine Anzahl von Frequenzkomponenten konzentrieren kann, wodurch sich Daten effizienter verarbeiten und komprimieren lassen.

Eigenschaften der DCT

- **Energiekompression**: Die DCT konzentriert die Energie eines Signals auf die niedrigfrequenten Komponenten, was sie ideal für die Komprimierung macht.
- **Orthogonalität**: Die Basisvektoren der DCT sind orthogonal, was eine einfache Rücktransformation ermöglicht.
- **Recheneffizienz**: Die DCT ist algorithmisch effizient und lässt sich schnell berechnen, ähnlich wie die schnelle Fourier-Transformation (FFT).
- **Verlustbehaftete Datenreduktion**: Durch das Entfernen hochfrequenter Komponenten können Daten mit minimalem Qualitätsverlust reduziert werden.
- **Anwendung**: Die DCT findet Anwendung in zahlreichen Bereichen, z. B. in der Bildkomprimierung (JPEG), Audioverarbeitung (MP3), und maschinellem Lernen.

Die DCT ist somit ein unverzichtbares Werkzeug für Informatiker, die mit der Verarbeitung und Komprimierung von Signalen und Bildern arbeiten. Sie verbindet mathematische Eleganz mit praktischer Effizienz und ist daher in vielen technischen Anwendungen unverzichtbar.

Eindimensionale DCT

Die eindimensionale DCT wird angewendet, um ein eindimensionales Signal, wie z. B. eine Audioaufnahme oder eine Reihe von Pixelwerten, zu analysieren. Die mathematische Definition lautet:

$$X_k = \alpha(k) \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}(n+0.5)k\right]$$

Hierbei gilt:

- x_n: Die Eingabedaten (z. B. Pixelwerte)
- X_k : Die transformierten Frequenzkomponenten
- N: Die Anzahl der Datenpunkte
- α(k): Eine Normierungsfunktion, um die Transformation zu skalieren

Die eindimensionale DCT zeichnet sich dadurch aus, dass sie vor allem die niedrigfrequenten Komponenten des Signals (z. B. die Grundstruktur eines Tons oder eines Bildes) hervorhebt, während hochfrequente Komponenten (Details oder Rauschen) weniger stark gewichtet werden.

(Ahmed, 1974) (Gonzalez, 2017) (Jain, 1989) (Khayam, 2003)

Zweidimensionale DCT

Die zweidimensionale DCT erweitert das Konzept der eindimensionalen DCT auf zweidimensionale Daten, wie beispielsweise Bilder. Die zweidimensionale DCT wird folgendermaßen definiert:

$$X_{k_1,k_2} = \alpha(k_1)\alpha(k_2) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{M-1} x_{n_1,n_2} \cos\left[\frac{\pi}{N}(n_1+0.5)k_1\right] \cos\left[\frac{\pi}{M}(n_2+0.5)k_2\right]$$

Hierbei gilt:

- $X_{n1,n2}$: Die Eingabematrix (z. B. Pixelmatrix eines Bildes)
- $X_{k1,k2}$: Die transformierte Frequenzmatrix
- N, M: Die Dimensionen der Eingabematrix

Die zweidimensionale DCT ist besonders nützlich, da sie die Energie eines Bildes auf eine kleine Anzahl von Frequenzkoeffizienten konzentrieren kann. Dies ermöglicht es, weniger wichtige Bildinformationen (meist hochfrequente Details) effizient zu entfernen oder stärker zu komprimieren, wie es z. B. im JPEG-Komprimierungsverfahren geschieht.

Funktionsweise

Die diskrete Kosinustransformation (DCT) funktioniert, indem sie Daten von der Zeit- oder Raumdomäne in den Frequenzbereich transformiert. Dadurch wird es möglich, die wesentlichen Informationen eines Signals oder Bildes auf eine kleine Anzahl von Frequenzkomponenten zu konzentrieren, was besonders für die Datenkomprimierung von Bedeutung ist.

Funktionsweise der eindimensionalen DCT

Die eindimensionale DCT wird auf Signale wie Audio oder eine Reihe von Pixelwerten angewendet. Sie berechnet für jede Frequenzkomponente einen Wert, der angibt, wie stark diese Frequenz im Signal vertreten ist. Niedrigfrequente Komponenten entsprechen dabei den Hauptstrukturen des Signals, während hochfrequente Komponenten oft Details oder Rauschen repräsentieren.

Beispiel: Betrachten wir ein Signal mit 8 Werten: [52, 55, 61, 66, 70, 61, 64, 73]. Die eindimensionale DCT transformiert dieses Signal in einen Frequenzbereich, wobei der erste Wert die DC-Komponente (Gesamthelligkeit) repräsentiert und die folgenden Werte die Amplituden der verschiedenen Frequenzkomponenten darstellen. Durch die Reduktion hochfrequenter Komponenten kann das Signal effizient gespeichert werden.

Funktionsweise der zweidimensionalen DCT

Die zweidimensionale DCT erweitert das Konzept auf zweidimensionale Daten, wie z. B. Bilder. Dabei wird die DCT auf jede Zeile und anschließend auf jede Spalte der Pixelmatrix angewendet. Das Ergebnis ist eine Frequenzmatrix, bei der die wichtigsten Bildinformationen in den unteren linken Bereich der Matrix konzentriert sind.

Beispiel: Ein Graustufenbild mit einer 8x8-Pixelmatrix wird durch die zweidimensionale DCT transformiert. Die resultierende Matrix enthält hohe Werte für die niedrigen Frequenzen (Hintergrund und große Muster) und geringe Werte für die hohen Frequenzen (feine Details). Indem man die hohen Frequenzen reduziert oder ignoriert, können die Bilddaten effizient komprimiert werden, wie es im JPEG-Standard geschieht.

Zusammenfassung

Die DCT ermöglicht es, die wesentlichen Informationen eines Signals oder Bildes kompakt darzustellen. Durch das Entfernen oder Reduzieren hochfrequenter Komponenten wird eine verlustbehaftete Komprimierung erreicht, die in der Praxis jedoch oft kaum wahrnehmbar ist. Dieses Verfahren findet Anwendung in der Bildkomprimierung, z. B. im JPEG-Format, und ist ein essenzielles Werkzeug für die effiziente Verarbeitung von Daten.

Praktischer Teil

Eindimensionale Kosinustransformation

Die diskrete Kosinustransformation (DCT) ist ein wichtiges Werkzeug in der Signal- und Bildverarbeitung, insbesondere für Anwendungen wie JPEG-Komprimierung. Dieser Code implementiert die eindimensionale DCT und ihre Rücktransformation (IDCT) in Python, um ein Signal in den Frequenzbereich zu transformieren und es anschließend wiederherzustellen.

Die Funktion *dct_transform* extrahiert Frequenzkomponenten eines Signals, während *idct_transform* diese wieder in das Originalsignal zurückführt. Ein Beispiel zeigt, wie die DCT zur Analyse und Komprimierung genutzt wird. Mit der Bibliothek *scipy.fftpack* wird eine effiziente und präzise Berechnung sichergestellt.

Funktion generate_dct_matrix

```python
def generate_dct_matrix(N):  1 usage
    """
    Erstellt die Transformationsmatrix für die eindimensionale diskrete Kosinustransformation (DCT).

    Parameters:
        N (int): Größe des Signals.

    Returns:
        numpy.ndarray: DCT-Transformationsmatrix.
    """
    dct_matrix = np.zeros((N, N))
    for k in range(N):
        for n in range(N):
            if k == 0:
                dct_matrix[k, n] = np.sqrt(1 / N)
            else:
                dct_matrix[k, n] = np.sqrt(2 / N) * np.cos(np.pi * k * (2 * n + 1) / (2 * N))
    return dct_matrix
```

Die Funktion *generate_dct_matrix(N)* erstellt die Transformationsmatrix für die eindimensionale diskrete Kosinustransformation (DCT) basierend auf der mathematischen Definition. Dabei werden die Einträge der Matrix entsprechend der DCT-Formel berechnet, wobei die erste Zeile normiert wird, um Gleichmäßigkeit zu gewährleisten. Diese Matrix wird verwendet, um ein Signal effizient in den Frequenzbereich zu transformieren.

Funktion dct_transform

```python
def dct_transform(signal):  1 usage
    """
    Führt die eindimensionale diskrete Kosinustransformation (DCT) eines Signals aus.

    Parameters:
        signal (numpy.ndarray): Eingabesignal (1D-Array).

    Returns:
        numpy.ndarray: Transformiertes Signal im Frequenzbereich.
    """
    return dct(signal, type=2, norm='ortho')
```

Die Funktion *dct_transform* führt die eindimensionale diskrete Kosinustransformation (DCT) eines Signals durch und wandelt es von der Zeit- in die Frequenzdomäne um. Dabei werden insbesondere niedrigfrequente Komponenten hervorgehoben, die die Hauptinformationen des Signals tragen. Das Ergebnis ist ein transformiertes Signal, das als Frequenzspektrum dargestellt wird.

Funktion idct_transform

```python
def idct_transform(transformed_signal):  1 usage
    """
    Führt die Rücktransformation (inverse DCT) eines transformierten Signals aus.

    Parameters:
        transformed_signal (numpy.ndarray): Signal im Frequenzbereich (1D-Array).

    Returns:
        numpy.ndarray: Rücktransformiertes Signal in der Zeitdomäne.
    """
    return idct(transformed_signal, type=2, norm='ortho')
```

Die Funktion *idct_transform* führt die Rücktransformation eines Signals im Frequenzbereich zurück in den Zeitbereich durch. Sie verwendet die inverse diskrete Kosinustransformation (IDCT) des Typs 2 und stellt sicher, dass das transformierte Signal orthonormalisiert ist. Damit ermöglicht sie die Rekonstruktion des ursprünglichen Signals basierend auf seinen Frequenzkomponenten.

Programmablauf

```python
# Beispiel: Demonstration der Funktion
if __name__ == "__main__":
    # Originalsignal
    signal = np.array([10, 20, 30, 40, 50, 60, 70, 80])
    print("Originalsignal:", signal)

    # DCT durchführen
    transformed_signal = dct_transform(signal)
    print("Transformiertes Signal (DCT):", transformed_signal)

    # Rücktransformation (IDCT) durchführen
    recovered_signal = idct_transform(transformed_signal)
    print("Rücktransformiertes Signal (IDCT):", recovered_signal)

    # Transformationsmatrix erstellen
    N = len(signal)
    dct_matrix = generate_dct_matrix(N)
    print("DCT-Transformationsmatrix:")
    print(dct_matrix)

    # DCT mit Transformationsmatrix durchführen
    transformed_with_matrix = np.dot(dct_matrix, signal)
    print("Transformiertes Signal mit Transformationsmatrix:", transformed_with_matrix)

    # Rücktransformation mit der Inversen der Transformationsmatrix durchführen
    inverse_dct_matrix = np.linalg.inv(dct_matrix)
    recovered_with_matrix = np.dot(inverse_dct_matrix, transformed_with_matrix)
    print("Rücktransformiertes Signal mit Inverser der Transformationsmatrix:", recovered_with_matrix)
```

Dieser Abschnitt demonstriert die Anwendung der diskreten Kosinustransformation (DCT) und ihrer Rücktransformation mithilfe der Transformationsmatrix. Zunächst wird die Transformationsmatrix für das gegebene Signal generiert und verwendet, um das Signal in den Frequenzbereich zu transformieren. Anschließend wird die Rücktransformation mit der Inversen der Transformationsmatrix durchgeführt, um das ursprüngliche Signal wiederherzustellen.

Output

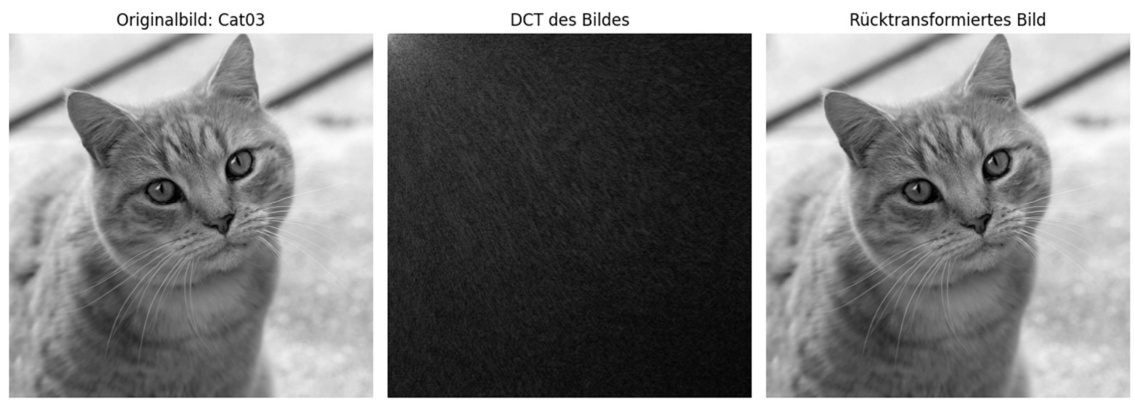
```
Originalsignal: [10 20 30 40 50 60 70 80]
Transformiertes Signal (DCT): [127.27922061 -64.42323023   0.          -6.73454801   0.          -2.00902904   0.          -0.50702323]
Rücktransformiertes Signal (IDCT): [10. 20. 30. 40. 50. 60. 70. 80.]
DCT-Transformationsmatrix:
[[ 0.35355339  0.35355339  0.35355339  0.35355339  0.35355339  0.35355339  0.35355339  0.35355339]
 [ 0.49039264  0.41573481  0.27778512  0.09754516 -0.09754516 -0.27778512 -0.41573481 -0.49039264]
 [ 0.46193977  0.19134172 -0.19134172 -0.46193977 -0.46193977 -0.19134172  0.19134172  0.46193977]
 [ 0.41573481 -0.09754516 -0.49039264 -0.27778512  0.27778512  0.49039264  0.09754516 -0.41573481]
 [ 0.35355339 -0.35355339 -0.35355339  0.35355339  0.35355339 -0.35355339 -0.35355339  0.35355339]
 [ 0.27778512 -0.49039264  0.09754516  0.41573481 -0.41573481 -0.09754516  0.49039264 -0.27778512]
 [ 0.19134172 -0.46193977  0.46193977 -0.19134172 -0.19134172  0.46193977 -0.46193977  0.19134172]
 [ 0.09754516 -0.27778512  0.41573481 -0.49039264  0.49039264 -0.41573481  0.27778512 -0.09754516]]
Transformiertes Signal mit Transformationsmatrix: [ 1.27279221e+02 -6.44232302e+01 -1.95399252e-14 -6.73454801e+00  3.55271368e-15 -2.00902904e+00 -1.77635684e-14 -5.07023228e-01]
Rücktransformiertes Signal mit Inverser der Transformationsmatrix: [10. 20. 30. 40. 50. 60. 70. 80.]
```

Der Output zeigt die erfolgreiche Transformation eines Signals mithilfe der diskreten Kosinustransformation (DCT) und der Rücktransformation (IDCT). Das Originalsignal [10, 20, 30, 40, 50, 60, 70, 80] wurde korrekt in den Frequenzbereich transformiert, und die Rücktransformation reproduziert das ursprüngliche Signal fehlerfrei. Die generierte DCT-Transformationsmatrix ermöglicht ebenfalls die Transformation und Rekonstruktion, was durch die nahezu identischen Ergebnisse mit minimalen numerischen Rundungsfehlern bestätigt wird.

Zweidimensionale Beispiele

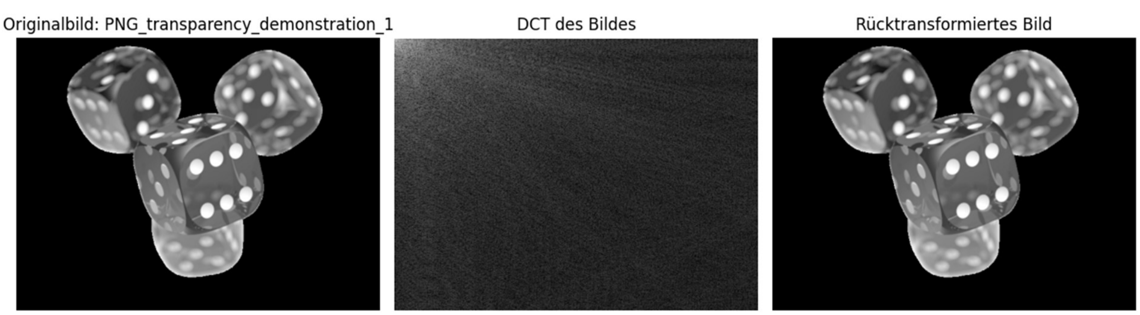
Das Skript veranschaulicht die praktische Nutzung der diskreten Kosinustransformation (DCT) zur Analyse und Verarbeitung von Bildern, darunter Fotos und Linienzeichnungen. Es zeigt die Originalbilder, transformiert diese in den Frequenzbereich und rekonstruiert sie anschließend in die Raumdomäne. Eine ausführliche Beschreibung würde den Rahmen sprengen, doch die Dokumentation der Funktionen macht die Funktionsweise im Quellcode nachvollziehbar. Dieses Beispiel bietet eine kompakte Demonstration der DCT und ihrer vielseitigen Einsatzmöglichkeiten.

JPG



Dieses Bild zeigt die Anwendung der diskreten Kosinustransformation (DCT) auf ein Foto einer Katze. Links ist das Originalbild zu sehen, in der Mitte die Darstellung des Bildes im Frequenzbereich nach der DCT, und rechts das rekonstruierte Bild nach der Rücktransformation (IDCT). Die Transformation und Rücktransformation zeigen, dass die Hauptmerkmale des Bildes trotz der Frequenzmanipulation erhalten bleiben.

PNG



Dieses Bild zeigt die Transformation eines Würfel-Renderings mithilfe der diskreten Kosinustransformation (DCT). Links ist das Originalbild, in der Mitte die Darstellung im Frequenzbereich nach der DCT und rechts das rekonstruierte Bild nach der Rücktransformation (IDCT). Die DCT ermöglicht es, die Frequenzkomponenten des Bildes zu analysieren, wobei die Rücktransformation zeigt, dass das ursprüngliche Bild nahezu verlustfrei wiederhergestellt werden kann.

Diskussion der Ergebnisse

Die eindimensionale DCT zeigt deutlich, wie Frequenzkomponenten eines Signals analysiert und komprimiert werden können. Dabei liegt der Fokus darauf, die Hauptinformationen des Signals, welche in den niedrigfrequenten Komponenten enthalten sind, effizient darzustellen. Dies ermöglicht eine präzise Analyse und Rücktransformation, wie es die Demonstration des Originalsignals zeigt.

Im zweidimensionalen Fall wird die DCT auf Bilddaten angewandt. Die Resultate zeigen, dass ähnliche Prinzipien gelten: Die niedrigen Frequenzkomponenten dominieren, und hochfrequente Details können für eine verlustbehaftete Komprimierung gezielt entfernt werden. Dies ist besonders bei Bildern wie dem Foto der Katze oder dem Würfel-Rendering zu beobachten, bei denen die Rücktransformation nahezu verlustfrei die Hauptmerkmale des Bildes wiederherstellt. Der Unterschied liegt vor allem in der Dimension der Daten, was die Komplexität der Berechnung und die Visualisierung der Frequenzbereiche beeinflusst.

Zusammengefasst zeigen beide Anwendungen der DCT ihre Stärke in der Datenreduktion und -rekonstruktion. Während die eindimensionale DCT ideal für Signalanalysen geeignet ist, bietet die zweidimensionale DCT eine effiziente Methode zur Bildkomprimierung und -analyse. Beide Methoden demonstrieren eindrucksvoll die Vielseitigkeit und Praktikabilität der diskreten Kosinustransformation in der modernen Datenverarbeitung.

Quellenverzeichnis

Ahmed, N. T. (1974). Discrete Cosine Transform. *IEEE Transactions on Computers*, 90-93.

Gonzalez, R. C. (2017). *Digital Image Processing.* London: Pearson.

Jain, A. K. (1989). *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall.

Khayam, S. A. (2003). *The Discrete Cosine Transform (DCT).* Michigan: Michigan State University.