Bildkompression mit Singulärwertzerlegung (SVD)

Maximilian Bazlov - Emmanuel Nana Nana

4. Januar 2026

Inhaltsverzeichnis

1 Einleitung

Die Kompression digitaler Bilddaten spielt eine zentrale Rolle im modernen Bildverarbeitungs- und Kommunikationsumfeld. Digitale Fotos und Grafiken bestehen typischerweise aus mehreren Millionen Pixeln, die jeweils durch einen oder mehrere Helligkeits- bzw. Farbkanäle beschrieben werden. Eine unkomprimierte Speicherung solcher Bilder führt daher leicht zu Dateigrößen im Bereich einiger Megabyte pro Bild. Dies ist sowohl für den lokalen Speicherbedarf als auch für die Übertragung über Netzwerke problematisch.

Um diesen Aufwand zu reduzieren, wurden zahlreiche Kompressionsverfahren entwickelt. In der Praxis ist insbesondere das JPEG-Format weit verbreitet, das auf der diskreten Kosinustransformation (DCT) basiert. Obwohl JPEG äußerst erfolgreich ist, ist es aus mathematischer Sicht interessant, alternative Ansätze zu betrachten, um die grundlegenden Prinzipien der Kompression besser zu verstehen.

In diesem Bericht untersuchen wir die *Singulärwertzerlegung* (SVD) als Werkzeug zur Bildkompression. Ein Graustufenbild wird durch eine Matrix $A \in \mathbb{R}^{m\times n}$ beschrieben. Die SVD zerlegt diese Matrix in

$$A = U\Sigma V^\top,$$

wobei U und V orthogonale Matrizen und Σ eine Diagonalmatrix mit Singulärwerten ist. Durch Beschränkung auf die k größten Singulärwerte erhält man eine Rang-k-Approximation A_k, die weniger Speicher benötigt.

Das Ziel ist es zu untersuchen, wie die Wahl von k den Kompressionsgrad und die Rekonstruktionsqualität beeinflusst. Der Satz von Eckart–Young–Mirsky garantiert dabei, dass die SVD die bestmögliche Approximation im Sinne der Frobeniusnorm liefert.

2 Mathematische Grundlagen

2.1 Singulärwertzerlegung von Matrizen

Sei $A \in \mathbb{R}^{m\times n}$ eine reelle Matrix. Die *Singulärwertzerlegung* (SVD) ist eine Faktorisierung

$$A = U\Sigma V^\top, \tag{1}$$

wobei $U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$ orthogonale Matrizen sind und Σ eine Diagonalmatrix mit Singulärwerten $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ ist. Die Spalten von U heißen *linke Singulärvektoren*, die Spalten von V *rechte Singulärvektoren*.

Die SVD kann als Summe von Rang-1-Matrizen geschrieben werden:

$$A = \sum_{i=1}^{r} \sigma_i\, u_i v_i^\top. \tag{2}$$

Der Term mit σ_1 trägt am stärksten zur Darstellung bei, die weiteren Terme beschreiben zunehmend feinere Strukturen. Diese Darstellung ist der Schlüssel zur Kompression: Indem man die Summation bei einem Index $k < r$ abbricht, erhält man eine Approximation mit reduziertem Rang.

Die Existenz der SVD kann über die Eigenwertzerlegung der symmetrischen Matrix $A^\top A$ begründet werden. Die Singulärwerte sind die Wurzeln der Eigenwerte von $A^\top A$, und die rechten Singulärvektoren sind die zugehörigen Eigenvektoren.

2.2 Rang-k-Approximation und Eckart–Young–Mirsky

Die Rang-k-Approximation von A ist definiert durch

$$A_k = \sum_{i=1}^{k} \sigma_i \, u_i v_i^\top = U_k \Sigma_k V_k^\top, \tag{3}$$

wobei U_k die ersten k Spalten von U, V_k die ersten k Spalten von V und Σ_k die Diagonalmatrix der größten k Singulärwerte enthält.

Satz (Eckart–Young–Mirsky): Unter allen Matrizen B mit $\operatorname{rang}(B) \leq k$ minimiert A_k den Approximationsfehler:

$$\|A - A_k\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2. \tag{4}$$

2.3 Anwendung auf Bilder

Ein Graustufenbild mit m Zeilen und n Spalten wird durch eine Matrix $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ repräsentiert, wobei $a_{ij} \in [0, 255]$ die Helligkeit darstellt. Der Wert 0 entspricht Schwarz, der Wert 255 entspricht Weiß.

Farbbilder mit drei Kanälen (RGB) werden als drei separate Matrizen $A^{(R)}, A^{(G)}, A^{(B)}$ modelliert. Die SVD wird auf jeden Kanal separat angewendet, und die rekonstruierten Kanäle werden anschließend wieder zu einem Farbbild zusammengesetzt. Dies ist möglich, da die Farbkanäle unabhängig voneinander sind.

2.4 Speicherbedarf und Kompressionsrate

Die Rang-k-Approximation $A_k = U_k \Sigma_k V_k^\top$ erfordert die Speicherung von:

- $m \cdot k$ Einträge für U_k
- k Einträge für die Diagonale von Σ_k
- $n \cdot k$ Einträge für $V_k^\top$

Insgesamt sind dies

$$N_k = k(m + n + 1) \tag{5}$$

reelle Zahlen pro Kanal. Bei Verwendung von 64-Bit-Gleitkommazahlen (8 Bytes) ergibt sich:

$$\text{Speicher}_k = k(m + n + 1) \cdot 8 \text{ Bytes}.$$

Die Kompressionsrate ist:

$$\text{Kompressionsrate} = \frac{m \cdot n}{k(m + n + 1)}. \tag{6}$$

Beispiel: Für ein 1000×1000 Bild:

k	N_k	Kompressionsrate	Anteil am Original
10	20.010	49,98	2,0%
50	100.050	9,99	10,0%
100	200.100	5,00	20,0%

3 Implementierung

3.1 Überblick über den Algorithmus

Der Ablauf der SVD-Kompression lässt sich in mehrere Schritte gliedern:

1. **Bildeinlesen:** Das Originalbild wird geladen und der Bildmodus bestimmt (Graustufen oder RGB). Bilder in anderen Modi werden automatisch konvertiert.
2. **Kanaltrennung:** Bei Farbbildern werden die drei Farbkanäle (Rot, Grün, Blau) als separate Matrizen extrahiert.
3. **SVD-Berechnung:** Für jede Matrix (bzw. jeden Kanal) wird die Singulärwertzerlegung berechnet und die Ergebnisse gespeichert.
4. **Rangwahl:** Der Benutzer wählt interaktiv den Rang k über einen Schieberegler.
5. **Approximation:** Für jeden Kanal wird die Rang-k-Approximation gebildet.
6. **Rekonstruktion:** Die approximierten Kanäle werden zusammengesetzt, und die Pixelwerte werden auf den gültigen Bereich $[0, 255]$ beschränkt.
7. **Ausgabe:** Das rekonstruierte Bild sowie Fehler- und Größenkennzahlen werden angezeigt.

3.2 Python-Umsetzung

Die Implementierung nutzt `NumPy`, `PIL` und `tkinter`. Die SVD-Berechnung erfolgt mit der optimierten Funktion aus NumPy:

```
U, s, V = numpy.linalg.svd(A, full_matrices=False)
```

Die Option `full_matrices=False` sorgt dafür, dass nur die relevanten Teile der Matrizen berechnet werden ("thin SVD"), was Speicher und Rechenzeit spart.

Die Rang-k-Approximation wird durch Auswahl der ersten k Komponenten gebildet:

```
A_k = (U[:, :k] @ numpy.diag(s[:k])) @ V[:k, :]
```

Der Approximationsfehler wird effizient aus den verworfenen Singulärwerten berechnet, ohne die Differenzmatrix explizit zu bilden:

```
fehler_norm = numpy.sqrt(numpy.sum(s[k:]**2))
```

Dies nutzt die Eigenschaft $\|A - A_k\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2$ und ist numerisch stabil.

3.3 Behandlung von Farbbildern

Bei RGB-Bildern wird die SVD auf jeden Kanal separat angewendet:

```
liste_U, liste_S, liste_V = [], [], []
for i in range(3):
    U, s, V = numpy.linalg.svd(data[:,:,i], full_matrices=False)
    liste_U.append(U); liste_S.append(s); liste_V.append(V)
```

Die rekonstruierten Kanäle werden mit `numpy.stack` zusammengesetzt und auf $[0, 255]$ beschränkt.

4 Benutzerhandbuch

4.1 Programmoberfläche

Das Programm bietet eine grafische Benutzeroberfläche (Abb. 1) mit:

- **Schaltfläche "Bild einlesen":** Öffnet einen Dateiauswahldialog.
- **Anzeigeflächen:** Original (links) und komprimiertes Bild (rechts).
- **Schieberegler:** Einstellung des Rangs k von 1 bis $\min(m, n)$.
- **Statusanzeigen:** Approximationsfehler (Frobeniusnorm) und Speichergröße.

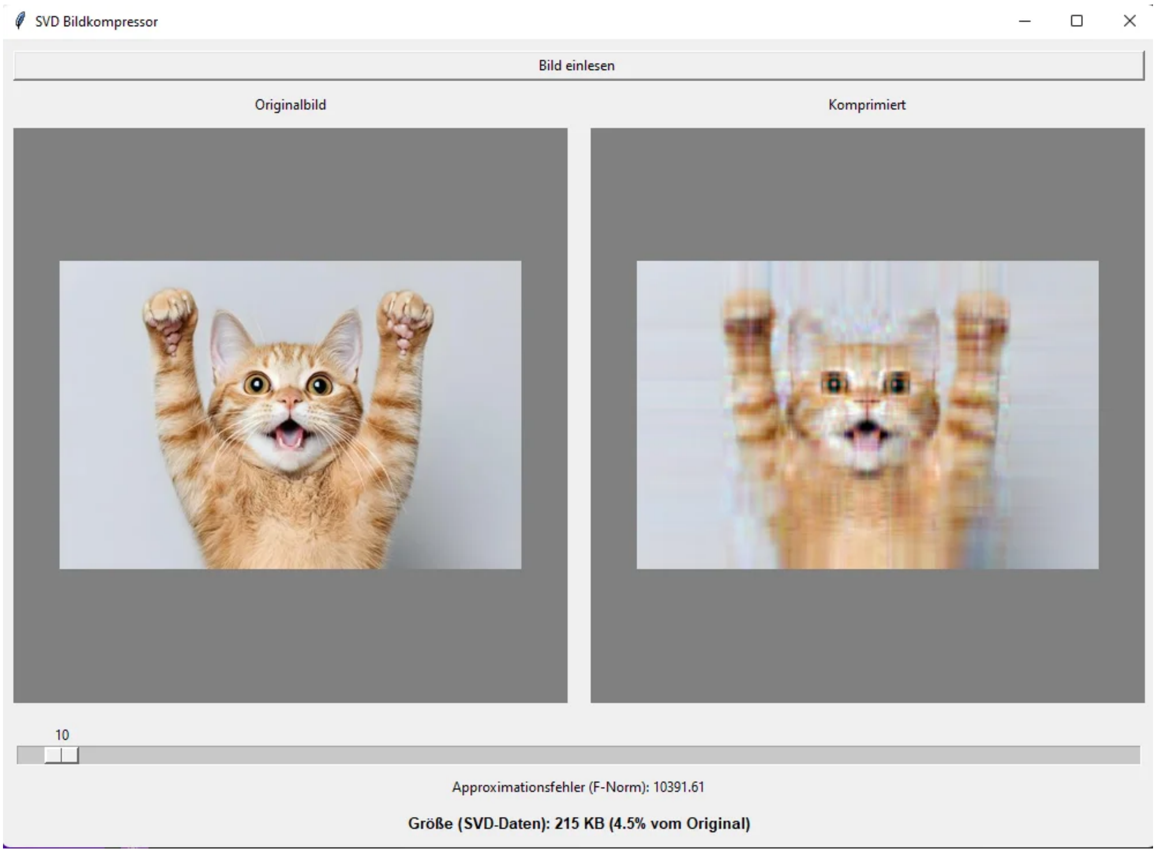

Abbildung 1: Benutzeroberfläche des SVD-Kompressors mit Original (links) und komprimiertem Bild (rechts). Der Schieberegler ermöglicht die interaktive Anpassung des Rangs k, während die Statusanzeigen den Approximationsfehler und die komprimierte Dateigröße anzeigen.

4.2 Arbeitsablauf

1. Auf "Bild einlesen" klicken und eine Bilddatei (JPG/PNG) auswählen.
2. Das Originalbild erscheint links, die SVD wird automatisch berechnet.
3. Mit dem Schieberegler den Rang k anpassen (Standardwert: 20).
4. Fehler und Größenangaben werden in Echtzeit aktualisiert.

4.3 Interpretation der Ausgaben

Approximationsfehler (F-Norm): Die Frobeniusnorm des Approximationsfehlers gibt an, wie stark das komprimierte Bild vom Original abweicht. Mathematisch ist dies $\|A - A_k\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}$. Kleinere Werte bedeuten bessere Approximation. Der Fehler nimmt mit steigendem k monoton ab und erreicht bei $k = \min(m, n)$ den Wert 0. Bei RGB-Bildern wird der Gesamtfehler über alle Kanäle summiert.

Größe (SVD-Daten): Diese Angabe zeigt den theoretischen Speicherbedarf der SVD-Faktoren U_k, Σ_k, V_k in Kilobyte sowie den prozentualen Anteil am Original. Die Berechnung basiert auf 64-Bit-Gleitkommazahlen (8 Bytes pro Zahl). Ein Wert von beispielsweise "10% vom Original" bedeutet, dass die SVD-Darstellung nur ein Zehntel des Speichers der Originalmatrix benötigt – also eine Kompression auf etwa ein Zehntel der ursprünglichen Größe.

4.4 Hinweise und Einschränkungen

- **Rechenzeit:** Die SVD-Berechnung erfolgt einmalig beim Laden und kann bei großen Bildern (mehrere Megapixel) einige Sekunden dauern. Die Komplexität ist $O(\min(m, n) \cdot m \cdot n)$.
- **Arbeitsspeicher:** Bei sehr großen Bildern kann die SVD-Berechnung mehrere hundert MB Arbeitsspeicher benötigen, da die vollständigen Matrizen U, Σ und V im Speicher gehalten werden.
- **Maximaler Rang:** Der Schieberegler ist auf $k \in [1, \min(m, n)]$ begrenzt.
- **Anzeigeskalierung:** Die angezeigten Bilder werden auf 400×400 Pixel skaliert, was nur der Darstellung dient und die Berechnungen nicht beeinflusst.
- **Farbmodi:** Bilder in anderen Farbmodi als RGB oder Graustufen (z. B. RGBA, CMYK) werden automatisch konvertiert.

5 Ergebnisse und Diskussion

5.1 Visuelle Demonstration

Die folgenden Abbildungen zeigen die Wirkung der SVD-Kompression bei verschiedenen Werten von k.

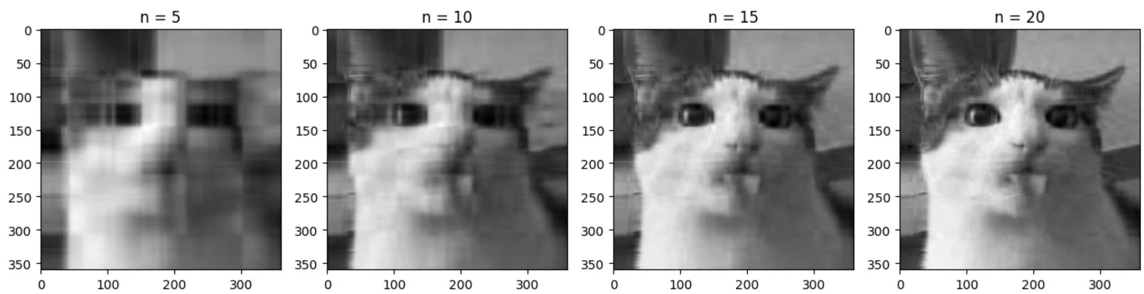

Abbildung 2: Rekonstruierte Bilder für verschiedene Rangwerte k ($n = 5$, $n = 10$, $n = 15$, $n = 20$). Mit steigendem k verbessert sich die Bildqualität deutlich sichtbar – von stark verschwommenen Konturen bei kleinem Rang bis hin zu gut erkennbaren Details.

5.2 Qualitative Beobachtungen

Die rekonstruierten Bilder zeigen charakteristische Eigenschaften in Abhängigkeit vom gewählten Rang k:

Sehr kleine k ($k < 10$): Das Bild wirkt stark verschwommen und weist deutliche Artefakte auf. Feine Details fehlen vollständig, aber grobe Strukturen wie dominante Farbflächen und die allgemeine Komposition sind bereits erkennbar. Bei $k = 1$ sieht man im Wesentlichen nur einen Gradienten in der Hauptrichtung der Bildvarianz.

Kleine bis mittlere k ($k = 10$–30): Konturen werden schärfer, und größere Strukturen sind gut erkennbar. Je nach Bildinhalt können erste Details wie Gesichtszüge oder Texturen sichtbar werden.

Mittlere k ($k = 30$–100): Die Bildqualität nähert sich dem Original merklich an. Feinere Details werden sichtbar, und das Bild wirkt natürlicher. Für viele Anwendungen ist die Qualität bereits akzeptabel.

Große k ($k > 100$): Der Unterschied zum Original ist nur bei genauer Betrachtung oder direktem Vergleich erkennbar. Feine Texturen werden zunehmend besser dargestellt.

5.3 Einfluss des Bildinhalts

Die Effektivität der SVD-Kompression hängt stark vom Bildinhalt ab:

- **Bilder mit glatten Farbverläufen:** Solche Bilder (z. B. Himmel, Wasser, computergenerierte Grafiken) haben typischerweise wenige dominante Singulärwerte und lassen sich sehr gut komprimieren. Bereits kleine k liefern gute Ergebnisse.
- **Bilder mit vielen feinen Details:** Detailreiche Bilder wie Landschaftsaufnahmen mit Vegetation oder Texturen haben ein flacheres Singulärwertspektrum. Hier werden höhere Werte von k für eine gute Approximation benötigt.
- **Bilder mit klaren Kanten:** Scharfe Kanten können zu wellenartigen Artefakten führen (ähnlich dem Gibbs-Phänomen bei Fourier-Approximationen), die bei niedrigem k sichtbar sein können.

5.4 Quantitative Analyse

Fehlerverhalten. Der Approximationsfehler $\|A - A_k\|_F$ nimmt mit steigendem k monoton ab. Die Abnahme ist anfangs schnell, da die ersten Singulärwerte typischerweise einen großen Teil der "Energie" des Bildes tragen. Ab einem gewissen Punkt flacht die Kurve ab, und zusätzliche Singulärwerte bringen nur noch marginale Verbesserungen.

Das Singulärwertspektrum in Abbildung 3 verdeutlicht diesen Zusammenhang: Die Singulärwerte fallen über mehrere Größenordnungen ab (von etwa 10^4 bis unter 10^{-10}), wobei der steilste Abfall in den ersten 50–100 Komponenten erfolgt.

Kompressionseffizienz. Die theoretische Kompression hängt linear von k ab. Für ein Bild der Größe $m \times n$ mit c Farbkanälen gilt:

$$\text{Komprimierte Größe} = k(m + n + 1) \cdot c \cdot 8 \text{ Bytes}.$$

Bei typischen Bildern ($m, n \approx 1000$) und kleinem k (z. B. $k = 50$) ergibt sich eine Reduktion auf etwa 10% der Originalgröße.

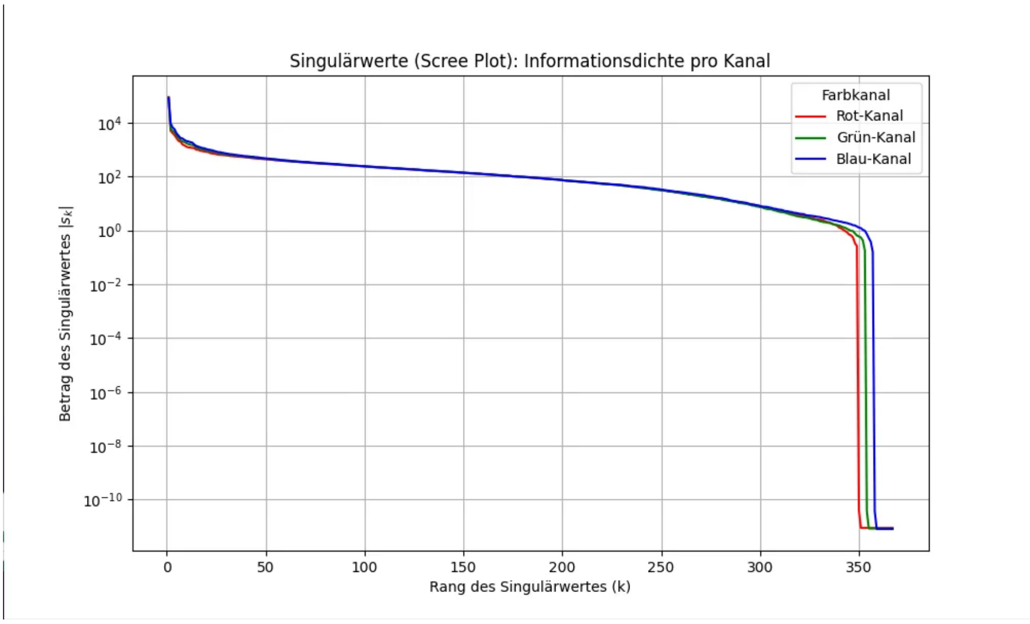

Abbildung 3: Singulärwertspektrum (Scree Plot) für die drei Farbkanäle (Rot, Grün, Blau). Die logarithmische Darstellung zeigt den typischen schnellen Abfall der Singulärwerte. Die ersten Singulärwerte tragen den Großteil der "Energie" des Bildes, während die späteren nur noch feine Details beschreiben.

5.5 Stärken und Schwächen der SVD-Kompression

Stärken:

- **Mathematische Fundierung:** Die Optimalitätseigenschaft nach Eckart–Young–Mirsky garantiert die bestmögliche Approximation für einen gegebenen Rang im Sinne der Frobeniusnorm.
- **Direkte Kontrolle:** Der Parameter k erlaubt eine intuitive und direkte Steuerung des Trade-offs zwischen Qualität und Kompression.
- **Keine Blockartefakte:** Im Gegensatz zu JPEG entstehen keine blockartigen Artefakte, da die SVD global auf der gesamten Bildmatrix arbeitet.
- **Glatte Degradation:** Die Bildqualität verschlechtert sich mit abnehmendem k gleichmäßig und vorhersehbar.
- **Theoretische Einsicht:** Die Methode verdeutlicht den Zusammenhang zwischen linearer Algebra und Bildverarbeitung.

Schwächen:

- **Rechenaufwand:** Die SVD-Berechnung hat eine Komplexität von $O(\min(m, n) \cdot m \cdot n)$ und ist damit für große Bilder rechenintensiv.
- **Praxiseffizienz:** In der Praxis erreicht die SVD-Kompression nicht die Effizienz spezialisierter Verfahren wie JPEG, die zusätzlich Wahrnehmungsmodelle nutzen.
- **Keine Wahrnehmungsoptimierung:** Die SVD minimiert den mathematischen Fehler, berücksichtigt aber nicht die Eigenschaften des menschlichen Sehens.

- **Globale Artefakte:** Bei niedrigem k erscheint das gesamte Bild verschwommen, statt dass lokale Details erhalten bleiben.

6 Schlussfolgerung und Ausblick

In diesem Bericht wurde die Singulärwertzerlegung (SVD) als Methode zur Bildkompression untersucht. Ausgehend von der Modellierung eines Bildes als Matrix erlaubt die SVD eine Zerlegung in orthogonale Matrizen und eine Diagonalmatrix mit Singulärwerten, die nach Wichtigkeit geordnet sind. Durch Beschränkung auf die k größten Singulärwerte entsteht eine Rang-k-Approximation, die gemäß dem Satz von Eckart–Young–Mirsky die bestmögliche Approximation im Sinne der Frobeniusnorm darstellt.

Das entwickelte Programm ermöglicht die interaktive Exploration der SVD-Kompression durch eine grafische Benutzeroberfläche. Der Benutzer kann ein Bild laden und mit einem Schieberegler den Rang k stufenlos variieren, wobei das komprimierte Bild sowie Fehler- und Größenangaben in Echtzeit aktualisiert werden.

Die Experimente zeigen, dass bereits relativ kleine Werte von k ausreichen können, um die wesentlichen Strukturen eines Bildes zu erhalten. Der Parameter k ermöglicht eine intuitive Kontrolle des Trade-offs zwischen Qualität und Kompression.

Die SVD-Kompression ist aus didaktischer Sicht besonders wertvoll, da sie den Zusammenhang zwischen Rangreduktion und Informationsgehalt klar veranschaulicht. Für praktische Anwendungen sind spezialisierte Verfahren wie JPEG jedoch effizienter, da sie zusätzlich Modelle der menschlichen Wahrnehmung und optimierte Codierung nutzen.

Mögliche Erweiterungen:

- **Zusätzliche Fehlermaße:** Implementierung von MSE (Mean Squared Error) und PSNR (Peak Signal-to-Noise Ratio) für bessere Vergleichbarkeit mit anderen Verfahren.
- **Speicherfunktion:** Möglichkeit, komprimierte Bilder als Datei zu speichern.
- **Singulärwertspektrum:** Grafische Darstellung der Singulärwerte zur Analyse der Bildstruktur.
- **Adaptive Wahl von k:** Automatische Bestimmung von k basierend auf einem vorgegebenen Energieanteil oder Qualitätskriterium.
- **Randomized SVD:** Einsatz approximativer Verfahren zur Beschleunigung bei sehr großen Bildern.
- **JPEG-Vergleich:** Direkte Gegenüberstellung mit JPEG bei gleicher Dateigröße.

Zusammenfassend ist die SVD ein wertvolles Werkzeug zum Verständnis der Bildkompression, das die grundlegende Idee der Rangreduktion anschaulich vermittelt. Die entwickelte Anwendung ermöglicht es, diese mathematischen Konzepte interaktiv zu erkunden und den Einfluss des Approximationsrangs auf die Bildqualität direkt zu beobachten.

Die praktische Erfahrung mit dem Programm zeigt, dass die optimale Wahl von k stark vom Bildinhalt abhängt. Während bei Bildern mit wenig Detailreichtum bereits $k \approx 20$–50 zu guten Ergebnissen führt, benötigen detailreiche Fotografien deutlich höhere Werte. Diese Beobachtung unterstreicht die zentrale Rolle des Singulärwertspektrums für die Komprimierbarkeit eines Bildes.