Projekt: Entrauschen eines Graustufenbildes

Prof. R. Hiptmair, SAM, D-MATH, ETH Zürich

Vorlesung “Lineare Algebra und Numerische Mathematik” (D-BAUG)

1 Einleitung

In Abschnitt 1.3.3 der Vorlesung haben wir gesehen, dass das Entrauschen eines Bildes, das durch lokale lineare Überlagerung von Pixelwerten gestört worden ist, durch das Lösen eines linearen Gleichungssystems erreicht werden kann.

In diesem Programmierprojekt wollen wir eine solche Entrauschprozedur in MATLAB implementieren und auf ein verrauschtes Bild anwenden. Dabei üben wir die Initialisierung *dünnbesetzter Matrizen* in MATLAB, siehe Abschnitt 5.4 der Vorlesung.

Es wird empfohlen, Abschnitt 1.3.3 der Vorlesung zu repetieren, bevor man die Aufgabe bearbeitet.

2 Bilder als Matrizen und Vektoren

Ein rechteckiges Graustufenbild mit m Zeilen und n Spalten wird repräsentiert durch eine $m \times n$-Matrix $\mathbf{P}$ von nichtnegativen Integerwerten zwischen 0 und einem Maximalwert. Handelt es sich etwa um ein Bild im BMP-Format mit 8 Bit Tiefe so ist der Maximalwert $2^8 - 1 = 255$. Der Grauwert des linken oberen Pixels des Bildes wird dabei im Matrixeintrag $(\mathbf{P})_{1,1}$ gespeichert, der rechte untere Pixelwert in $(\mathbf{P})_{m,n}$, siehe Abbildung 1. Für weitere Erklärungen wird auf Abschnitt 1.3.3 der Vorlesung verwiesen.

Die MATLAB-Funktion `imread` liest ein Bild ein, erkennt das Format und wandelt es in eine oder mehrere Integermatrizen um. In diesem Projekt werden wir jedoch den ganzzahligen Datentyp ignorieren und annehmen, dass wir mit Matrizen mit reellen Einträgen rechnen. Mit Hilfe der MATLAB-Funktion `image` kann das Bild dargestellt werden, siehe Listing 1.

Listing 1: Einlesen und Darstellen eines Bildes im BMP-Format

```
1 function P = readblurredimg(filename)
2 % MATLAB function for reading a monochrome BMP image into a matrix
3 if (nargin < 1), filename = 'lanmblurred.bmp'; end
4 P = double(imread(filename));
5 % print size and display image
6 [no_rows,no_cols] = size(P),
```

```
figure; colormap(gray); image(uint8(P));
```

$$\mathbf{P} = \begin{bmatrix} 158 & 247 & 11 & 248 & 189 \\ 155 & 48 & 27 & 90 & 37 \\ 243 & 156 & 85 & 25 & 187 \\ 130 & 201 & 102 & 97 & 152 \\ 65 & 169 & 94 & 239 & 60 \end{bmatrix}$$

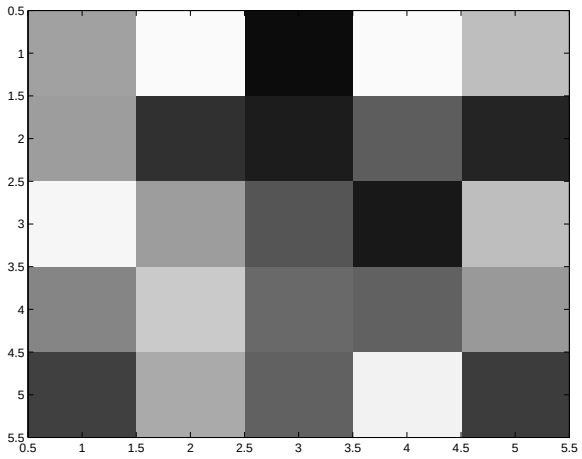
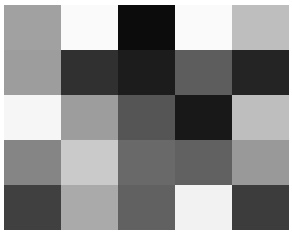


Abbildung 1: Graustufenpixelbild codiert in einer Matrix $\mathbf{P} \in \mathbb{R}^{5,5}$

Wie in Abschnitt 1.3.3 der Vorlesung erklärt können, die Bilddaten auch als Vektor $\mathbf{p} \in \mathbb{R}^{m \cdot n}$ interpretiert werden, sobald man sich auf eine Anordnung der Pixel verständigt. Eine Möglichkeit ist die zeilenweise lexikographische Anordnung mit der Identifikation

$$(\mathbf{p})_\ell \quad \Longleftrightarrow \quad (\mathbf{P})_{i,j} \quad \text{, falls} \quad \ell = n * (i-1) + j \;, \tag{1}$$

die etwa für ein 5×5-Bild auf folgende Numerierung führt:



$\longleftrightarrow$

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Aufgabe. Schreiben Sie eine MATLAB-Funktion

```
function pvec = makeimgvector(Pmat) ,
```

die ein Bild, gespeichert in einer Matrix $\mathbf{P} \in \mathbb{R}^{m,n}$ (Argument `Pmat`) nimmt und den entsprechenden Spaltenvektor $\mathbf{p} \in \mathbb{R}^{mn}$ gemäss (1) zurückgibt.

Hinweis: Sie könne die MATLAB-Funktion `reshape` benutzen oder auch die Tatsache, dass für eine Matrix `A` der Befehl `A(:)` diese in einen Vektor umwandelt (der allerdings noch nicht der richtige ist!).

3 Verrauschen eines Bildes

Gegeben sei ein Graustufenbild mit m Zeilen und n Spalten in der Matrix $\mathbf{P} \in \mathbb{R}^{m,n}$ mit $0 \leq (\mathbf{P})_{i,j} \leq p_{\max}$. Durch lokale lineare Überlagerung von Pixelwerten wird es bei der Übertragung

gestört und man empfängt das *verrauschte* Bild $\widetilde{\mathbf{P}} \in \mathbb{R}^{m,n}$. Dessen Pixelwerte sind gegeben durch

$$\left(\widetilde{\mathbf{P}}\right)_{i,j} = \sum_{k=-p}^{p} \sum_{\ell=-p}^{p} s_{k+p+1,\ell+p+1} (\mathbf{P})_{i+k,j+\ell} \ , \quad i \in \{1,\ldots,m\},\ j \in \{1,\ldots,n\} \ , \tag{2}$$

für bekannte Faktoren $s_{k,\ell} \in \mathbb{R}$, $k,\ell \in \{1,\ldots,2p+1\}$, $p \in \mathbb{N}_0$, siehe Formel (I.3.F) aus der Vorlesung. Diese Faktoren werden zu einer Matrix $\mathbf{S} \in \mathbb{R}^{2p+1,2p+1}$ mit $(\mathbf{S})_{k,\ell} := s_{k,\ell}$ zusammengefasst, die als *Point Spread Function* (PSF) bezeichnet wird. Die PSF-Faktoren müssen erfüllen:

$$s_{k,\ell} \geq 0 \quad \text{und} \quad \sum_{k=1}^{2p+1} \sum_{\ell=1}^{2p+1} s_{k,\ell} = 1 \ . \tag{3}$$

Dies garantiert, dass $0 \leq \left(\widetilde{\mathbf{P}}\right)_{i,j} \leq p_{\max}$ (Warum?), so dass auch $\widetilde{\mathbf{P}}$ wieder ein Bild vom gleichen Format beschreibt.

Die Formel (2) ist noch nicht wohldefiniert, denn die Indices in $(\mathbf{P})_{i+k,j+\ell}$ können aus dem erlaubten Bereich fallen! Man muss daher die folgende **Konvention** vereinbaren: Terme in (2), für die

$$i+k \notin \{1,\ldots,m\} \quad \text{oder} \quad j+\ell \notin \{1,\ldots,n\}$$

werden bei der Summation nicht berücksichtigt.

Wir verwenden im Folgenden die PSF

$$\mathbf{S} := \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.01 \\ 0.01 & 0.02 & 0.03 & 0.03 & 0.03 & 0.02 & 0.01 \\ 0.01 & 0.02 & 0.03 & 0.20 & 0.03 & 0.02 & 0.01 \\ 0.01 & 0.02 & 0.03 & 0.03 & 0.03 & 0.02 & 0.01 \\ 0.01 & 0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \end{bmatrix} \in \mathbb{R}^{7,7} \ , \tag{4}$$

also $p = 3$. Die Wirkung von (2) mit dieser PSF auf ein Bild ist illustriert in Abbildung 2.

Aufgabe. Schreiben Sie eine MATLAB-Funktion

```
function Pblurred = psfblur(P,S) ,
```

die aus dem Graustufenbild, das in der Matrix $\mathbf{P}$ (Argument `P`) gespeichert ist, und der PSF, übergeben in der quadratischen Matrix `S`, das gemäss (4) verrauschte Bild berechnet.

Hinweis. Hier dürfen Sie ohne Bedenken geschachtelte Schleifen und geeignete Abfragen verwenden.

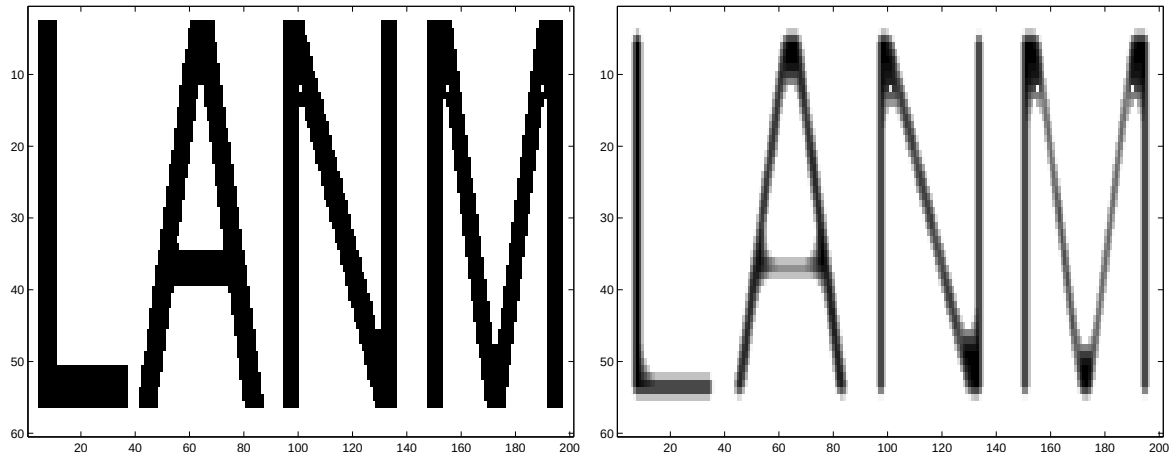


Abbildung 2: Originalbild (links) und verrauschtes Bild (rechts), PSF gegeben durch (4)

4 Entrauschen

Die entscheidende Einsicht ist, dass

> sich die Transformation $\mathbf{P} \to \widetilde{\mathbf{P}}$ gemäss (4) durch Multiplikation des $\mathbf{P}$ gemäss (1) zugeordneten Vektors mit einer *dünnbesetzten* $mn \times mn$*-Matrix* mit einer speziellen Struktur realisieren lässt.

Für $m = n = 20$ und die PSF aus (4) ist das Besetzungsmuster dieser Matrix in Abbildung 3 als "`spy`-Plot" dargestellt. Diese "Verrauschungsmatrix" lässt sich jeweils aus $m \cdot n$ kleinen Bandmatrizen zusammensetzen.

Aufgabe. Skizzieren Sie das Besetzungsmuster der "Verrauschungsmatrix" für $m = n = 10$ und $p = 0$ und $p = 1$.

Hauptaufgabe. Schreiben Sie eine MATLAB-Funktion

```
function A = buildpsfmatrix(m,n,S),
```

die die $mn \times mn$ "Verrauschungsmatrix" für ein Bild mit m Zeilen und n Spalten als dünnbesetzte Matrix initialisiert, wobei `S` die PSF übergibt.

Hinweis. Absolute MATLAB-Profis erledigen das ohne eine einzige Schleife, Normalsterbliche initialisieren die Matrix Zeile für Zeile mit Hilfe von zwei Schleifen über alle Pixel. Als Zwischendatenstruktur sollte das Indexpaar-Eintrag-Format verwendet werden, das die `sparse`-Funktion von MATLAB verwendet.

Hat man diese Matrix erst einmal erstellt, ist das Entrauschen ganz einfach — na ja, relativ einfach, siehe Listing 2.

Listing 2: Entrauschen eines Graustufenbildes mittels Lösen eines dünnbesetzten linearen Gleichungsssystems

```
1 function Psharp = deblur(Pblur,S)
```

```
% MATLAB function for deblurring via solving a sparse linear system
[m,n] = size(Pblur);
v = buildpsfmatrix(m,n,S)\makeimgvector(Pblur);
Psharp = reshape(v,n,m)';
```

Aufgabe. Wenden Sie Ihre MATLAB-Codes an, um das Bild in der Datei `lanmblurred.bmp`, die im Projektordner zur Verfügung gestellt wird, zu entrauschen. Die Störung wurde durch die PSF aus (4) bewirkt.

Bemerkung. Dank der speziellen Struktur der "Verrauschungsmatrix" lassen sich lineare Gleichungssysteme mit ihr als Koeffizientenmatrix mit Hilfe einer speziellen Methode, der *diskreten Fouriertransformation* lösen. Diese Methode ist implementiert in der MATLAB-Funktion `fft` und ist noch wesentlich schneller als \.

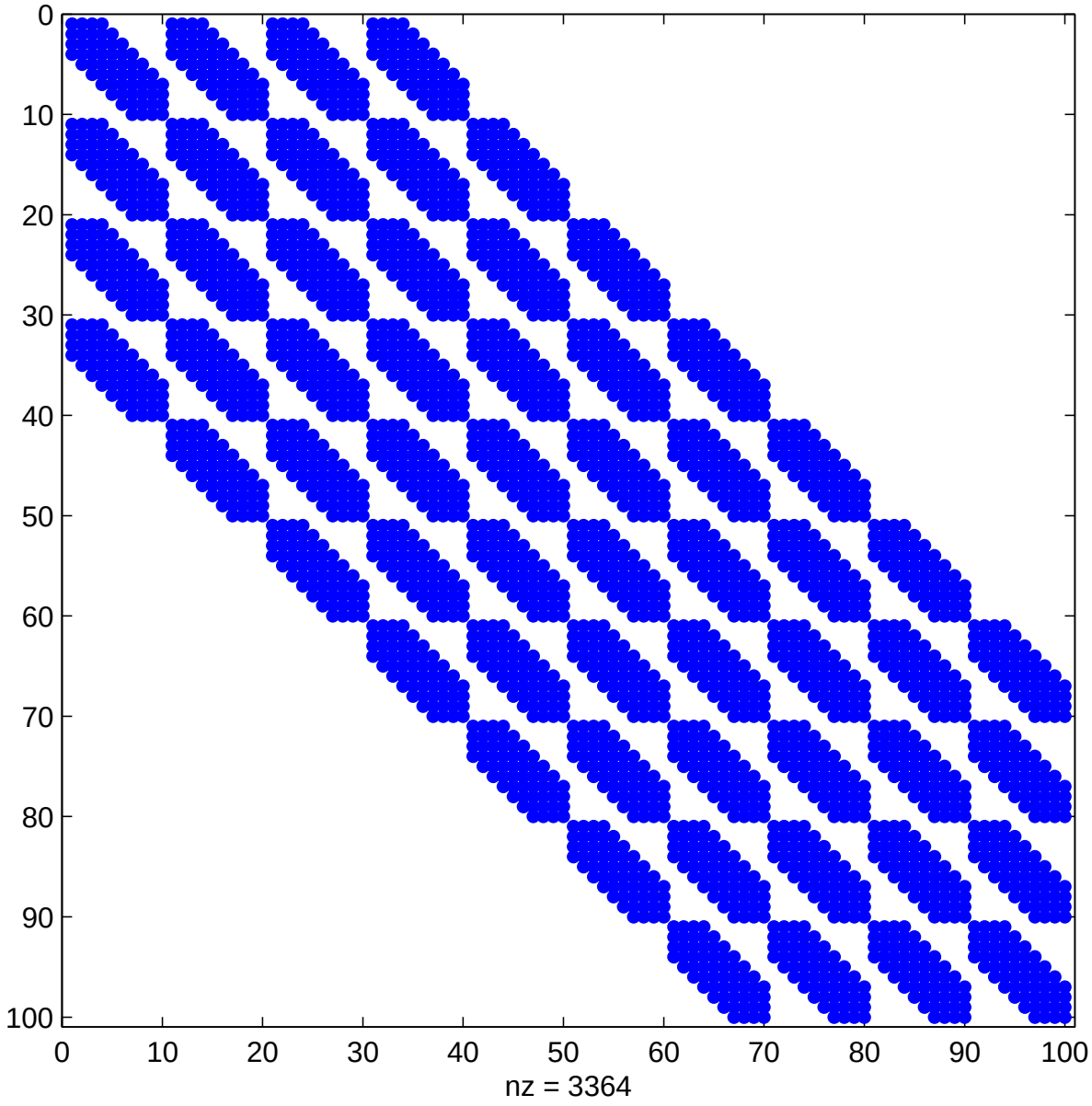


Abbildung 3: Besetzungsmuster der dünnbesetzten Verrauschungsmatrix für $p = 3$, $m = n = 20$

Listing 3: Initialisierung der Verrauschungsmatrix

```
function A = buildpsfmatrix(m,n,S)
% MATLAB function building the matrix for the linear transformation
    corresponding to a blurring of an gray scale image with m×n
    pixels by means of a point spread function encoded in the matrix
    S
[p,q] = size(S); % obtain number of rows and columns of S
if ((mod(p,2) == 0) || (mod(q,2) == 0)), error('S must have odd no
    of rows/columns'); end
p = (p-1)/2; q = (q-1)/2; % spread of psf in both diretions

% Initialize sparse matrix through index-value triplets
% Numbering assumes lexikographic top->bottom ordering of pixels
i_glob = []; j_glob = []; a_glob = []; % Arrays for assembling the
    matrix
% Run through all the pixels and encode one row of the blurring
    transform
for i=1:m
  for j=1:n
    % Focus on stencil centered at pixel (i,j)
    i_row = (i-1)*n + j; % index of matrix row corresponding to
       pixel (i,j)
    % The MATLAB command meshgrid creates two matrices of components
       of index pairs corresponding to the entries of a matrix
    [i_ofs,j_ofs] = meshgrid(-p:p,-q:q);
    % position indices of pixels affected by PSF centered at (i,j)
    i_idx = i + i_ofs(:); j_idx = j + j_ofs(:);
    % Next, remove out-of-bound indices
    in_bound = find((i_idx > 0) .* (i_idx <= m) .* (j_idx > 0) .*
       (j_idx <= n));
    i_idx = i_idx(in_bound); j_idx = j_idx(in_bound);
    s_in = S(in_bound);
    % Augment global arrays describing sparse matrix
    i_glob = [i_glob;i_row*ones(numel(i_idx),1)]; % row with index
       i_row
    j_glob = [j_glob;n*(i_idx-1)+j_idx]; % column indices
    a_glob = [a_glob;s_in];
  end
end
% Assemble sparse matrix
A = sparse(i_glob,j_glob,a_glob,m*n,m*n);
```