1 Digital images as vectors in $\mathbb{R}^n$

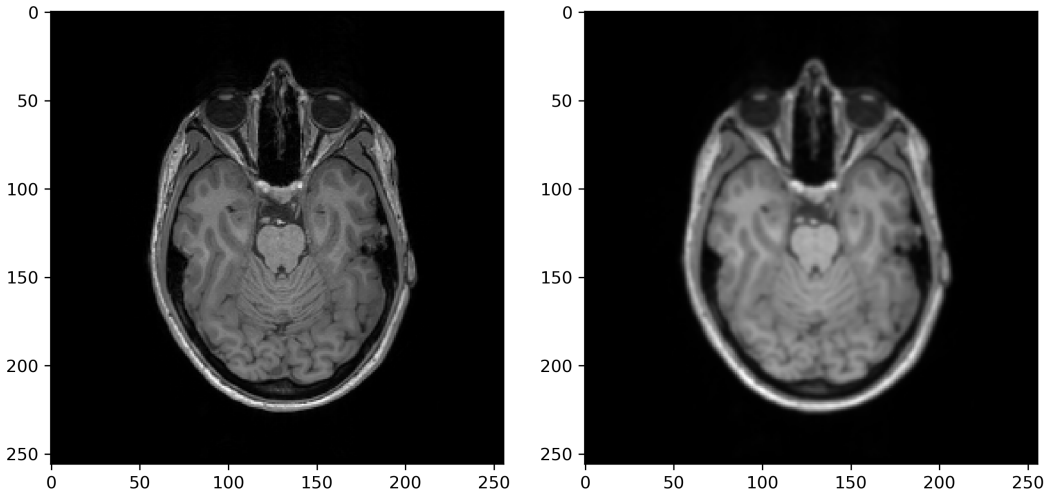
Throughout this workshop we will work with very simple image processing models. We will focus on grayscale images only. In our model, we will think of a $N \times M$ image as a collection of $n = NM$ real numbers representing the grayscale levels of individual pixels.[1] This way we can also think of such images as vectors in $\mathbb{R}^n$. Throughout this document we will accept the following way of numbering pixels in images:

2×3 grayscale image: $\begin{array}{|c|c|c|}\hline P_{1,1} & P_{1,2} & P_{1,3} \\ \hline P_{2,1} & P_{2,2} & P_{2,3} \\ \hline \end{array}$ $\quad\Longleftrightarrow\quad$ vector $\begin{bmatrix} P_{1,1} \\ P_{1,2} \\ P_{1,3} \\ P_{2,1} \\ P_{2,2} \\ P_{2,3} \end{bmatrix} \in \mathbb{R}^6$

In this way the grayscale value of pixel (i, j), $i = 1, \ldots, N$, $j = 1, \ldots, M$ corresponds to the $M(i-1)+j$ component in the equivalent vector representation. We will use double index notation to refer to the pixels in the image, and single index notation to refer to the components in the vector.

2 Blurring of images

Because of optical aberrations, we will assume that instead of some original image $N \times M$ image O, we record its blurred version B. For example:

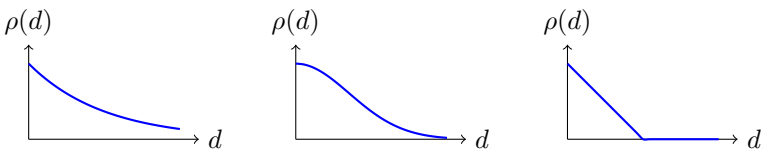


Original image O Blurred image B

We will model this process as follows. It is natural to assume that each pixel $B_{i,j}$ in the blurred image is a weighted average of pixels of the original image O around (i, j). More precisely, given a "weighting" function $\rho : \mathbb{R} \to \mathbb{R}$, the following model holds:

$$B_{i,j} = \sum_{k=1}^{N} \sum_{\ell=1}^{M} \rho\left(d_{ijk\ell}\right) O_{k,\ell}, \qquad i = 1, \ldots, N; j = 1, \ldots, M, \tag{1}$$

where $d_{ijk\ell} = \sqrt{(i-k)^2 + (j-\ell)^2}$ is the distance between pixels (i, j) and (k, ℓ). The most natural situation arises when ρ is decreasing for increasing distances between pixels, so that pixels, which are further away from (i, j), contribute with smaller weight to the blurred result; for example:



$\rho(d) = \exp\{-d\}$ $\rho(d) = \exp\{-d^2\}$ $\rho(d) = \max\{1-d, 0\}$

[1] That is, we will ignore the fact that digital images contain only a discrete number of grayscale levels.

We will use the symbol $T : \mathbb{R}^n \to \mathbb{R}^n$ to denote the blurring process applied to our vector representation of images. That is, $T(O) = B$ means that images O and B are related through (1).

Question 1.

1.1. Verify that the function T is linear.

1.2. Let ρ, N, and M be given. Determine the expression for the elements of the standard matrix A for T.

1.3. Suppose that there is a number $\hat{d} > 0$ such that $\rho(d) = 0$ for $d > \hat{d}$. Explain why some elements in the standard matrix A for T can be expected to be 0 in this case.

3 Deblurring of images

Our goal now is to recover the original image O given the blurred image B and the knowledge of the blurring process (1).

Invertibility of the blurring transformation

Our first order of business is to find out whether recovering the original image O given a blurred image B is even possible. That is, we would like to know, when the blurring transformation is *invertible*. Recall that a linear transformation is invertible $\iff$ its standard matrix is invertible $\iff$ the homogeneous system of linear equations admits only a trivial solution.

We will now give a very simple sufficient criterion for invertibility.

A square matrix $n \times n$ A is called *strictly diagonally dominant*, if in each row the diagonal element is larger than the sum of absolute values of off-diagonal elements, that is:

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}|, \quad \forall i = 1, \ldots, n.$$

Theorem 1. *Strictly diagonally dominant matrices are invertible.*

Proof. Indeed, assume that A is a strictly diagonally dominant matrix, and $Ax = 0$ for some $x \in \mathbb{R}^n$. Suppose that $x \neq 0$, and let the index $\hat{i}$ be the index with largest (in absolute magnitude) component of x, that is:

$$|x_{\hat{i}}| \geq |x_i|, \quad \forall i = 1, \ldots, n.$$

Note that by our assumption $x \neq 0$ and consequently $|x_{\hat{i}}| > 0$. Let us consider equation number $\hat{i}$ in our homogeneous system $Ax = 0$:

$$A_{\hat{i},1}x_1 + A_{\hat{i},2}x_2 + \cdots + A_{\hat{i},n}x_n = 0,$$

or

$$A_{\hat{i},\hat{i}}x_{\hat{i}} = -\sum_{j \neq \hat{i}} A_{\hat{i},j}x_j.$$

We take the absolute value of both sides and divide the result by $|x_{\hat{i}}| > 0$:

$$|A_{\hat{i},\hat{i}}| = \frac{\left|\sum_{j \neq \hat{i}} A_{\hat{i},j}x_j\right|}{|x_{\hat{i}}|} \leq \sum_{j \neq \hat{i}} |A_{\hat{i},j}| \underbrace{\frac{|x_j|}{|x_{\hat{i}}|}}_{\leq 1} \leq \sum_{j \neq \hat{i}} |A_{\hat{i},j}|,$$

where we used triangle inequality for the absolute value, and the fact that $\frac{|x_j|}{|x_{\hat{i}}|} \leq 1$ because of our selection of the index $\hat{i}$. Note that we arrived at the inequality that violates the assumption of strict diagonal dominance of A! Therefore, our assumption that $x \neq 0$, which implied that $|x_{\hat{i}}| > 0$, is wrong and the homogeneous system admits only a trivial solution. In other words, the matrix A is invertible. ∎

Question 2. The standard matrix for the blurring transformation (1) is strictly diagonally dominant if the "weighting" function ρ satisfies the following vaguely formulated conditions:

1. $|\rho(0)| > 0$.
2. $|\rho(d)|$ becomes small, relative to $|\rho(0)|$, with increasing distance d.

Discuss the relation of these conditions to the definition of strictly diagonally dominant matrices. If possible, try to formulate them more precisely in your report.

Iterative deblurring algorithm

Note that the size of the linear system (number of equations and unknowns) corresponding to the image blurring process is proportional to the number of pixels in the image. Therefore, even for a meager 100×100 image we need to solve a system with 10000 equations and unknowns to deblur the image. Therefore, we will attempt to utilize a simple approximate, iterative approach - in the same spirit as the power method for eigenvalue/eigenvector computation.

Given a strictly diagonally dominant $n \times n$ matrix A and a right hand side $b \in \mathbb{R}^n$ (this will correspond to a blurred image in our case), we will construct a sequence of approximate solutions $x^{(k)}$, $k = 0, 1, 2, \ldots$ to the system $Ax = b$ as follows.

1: Select an arbitrary $x^{(0)} \in \mathbb{R}^n$
2: **for** $k = 0, 1, 2 \ldots$ **do**
3: **for** $i = 1, 2, \ldots, n$ **do**
4: Solve equation i independently from all other equations:

$$x_i^{(k+1)} = A_{ii}^{-1}\left(b_i - \sum_{j \neq i} A_{ij} x_j^{(k)}\right)$$

5: **end for**
6: **if** $\|x^{(k+1)} - x^{(k)}\|$ is "small enough" **then**
7: Stop, $x^{(k+1)}$ is an approximate solution.
8: **end if**
9: **end for**

One can show that this algorithm, known as the Jacobi iteration, converges for strictly diagonally dominant matrices to the solution of the linear system $Ax = b$ from an arbitrary initial guess $x^{(0)}$.

Question 3.

3.1. Assume that at some iteration of the algorithm above we have $x^{(k+1)} = x^{(k)}$ as exact equality. Show that $Ax^{(k+1)} = b$, that is, $x^{(k+1)}$ is the solution to our system of linear equations. This provides some motivation for the stopping criterion in the algorithm.

3.2. Show that the update from $x^{(k)}$ to $x^{(k+1)}$ (lines 3–5 in the algorithm above) can be equivalently implemented as follows:

$r^{(k)} = b - Ax^{(k)}$
for $i = 1, 2, \ldots, n$ **do**
$x_i^{(k+1)} = x_i^{(k)} + A_{ii}^{-1} r_i^{(k)}$
end for

3.3. In the context of image deblurring, what would be a reasonable choice for the initial solution guess $x^{(0)}$ in this iterative algorithm?

4 Implementation exercises

On Moodle you will find an example script that reads, modifies, and saves PNG images (the script relies on `pngjs`), as well as a range of tiny test images.

Question 4.

4.1. Implement a function, which performs blurring in accordance with (1). That is, given a weighting function ρ, the function should take in an $N \times M$ image O represented as an array of length $n = NM$, and output a blurred image B, represented as an array of length $n = NM$.

Experiment with different weighting functions (for example, you can use Gaussian function $\rho(d) = \exp(-d^2/\sigma^2)$ and vary the "dispersion" parameter $\sigma > 0$).

4.2. Implement the iterative deblurring algorithm. Note that the version of the algorithm discussed in Question 3.2 can be implemented with the help of the "blurring" function. Indeed the matrix-vector product Ax corresponds to the application of the blurring transformation to x.

To verify the correctness of your implementation, start with an original image $\hat{O}$ and apply blurring transformation to it to get a blurred image B. Then one can expect that the sequence of approximations $O^{(k)}$ to the original image, which is produced by the iterative deblurring algorithm, converges to $\hat{O}$ with increasing number of iterations k **provided that the matrix for the linear transformation is strictly diagonally dominant**!

Hints:

- When debugging your algorithm, use tiny images, e.g. 8×8
- You are free to experiment with different weighing functions, but I suggest using Gaussian weighing function $\rho(d) = \exp(-d^2/\sigma^2)$, where $\sigma > 0$ is a parameter.
- For the purposes of deblurring it is important to make sure that the weighting function that you use results in a strictly diagonally dominant matrix for the blurring transformation. You can test this in your program by inspecting each row of the matrix and checking, whether the definition holds. If you use the Gaussian weighing function, by decreasing σ you should be able to make the standard matrix for the blurring transformation more diagonally dominant, see Question 2.