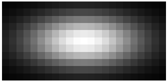
More general image blurs

Typical example Convolutional blur. Each pixel intensity value is 'spread out' to the neighbouring ones according to a (constant) point spread function matrix, e.g.,

$$\frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & \underline{1} & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \frac{1}{13}\begin{bmatrix} 0.2 & 0.4 & 0.6 & 0.4 & 0.2 \\ 0.4 & 0.6 & 0.8 & 0.6 & 0.4 \\ 0.6 & 0.8 & \underline{1} & 0.8 & 0.6 \\ 0.4 & 0.6 & 0.8 & 0.6 & 0.4 \\ 0.2 & 0.4 & 0.6 & 0.4 & 0.2 \end{bmatrix},$$



```
>> X = imread('cameraman.tif'); %standard test image
>> P = gaussianblur(5,11); %(not a standard function)
>> B = conv2(X, P, 'same');
>> imshow(uint8(B))
```

Can we undo this transformation?

Boundary conditions

Another point we need to specify: what do we do on the borders, with points where the blurring mask 'spills out'?
Natural choices:

- Ignore the contributions from pixels outside the image (i.e., set them to zero). Gives slightly blacker border.
- Repeat the image periodically: makes sense for images with a uniform background, e.g., a black one. (periodic boundary conditions).
- Mirror the image at the border (reflective boundary conditions).
- Other ad-hoc modifications, for instance estrapolate, or take the 'last known pixel' in each direction. Slightly more problematic in terms of the matrix structures they will involve.

Periodic B.C. + Known PSF is a setup that makes sense for astronomical images, e.g., `imshow('m83.tif')`.

What does the matrix look like? (1D)

First of all: the 1D version: convolution / moving averages.
Convolving each 'signal element' with $[p_{-k}, \ldots, p_{-1}, p_0, p_1, \ldots, p_k]$ gives a Toeplitz matrix (apart from the boundary conditions, at least).

$$A = \begin{bmatrix} p_0 & p_{-1} & \cdots & p_{-k} & & b.c. \\ p_1 & p_0 & p_{-1} & \cdots & p_{-k} & \\ p_2 & \ddots & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & & \\ p_k & \ddots & \ddots & \ddots & & \\ & \ddots & & & & \\ & & \ddots & & & \\ b.c. & & & & & \end{bmatrix}$$

zero boundary conditions $\implies$ band Toeplitz matrix
periodic boundary conditions $\implies$ circulant matrix
$A_{ij} = p_{mod(j-i,n)}$: constant along 'broken diagonals'.

What does the matrix look like? (2D)

$$\begin{bmatrix} T_0 & T_1 & \dots & T_k & & b.c. \\ T_{-1} & T_0 & T_1 & \dots & T_k & \\ T_{-2} & \ddots & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & & \\ p_{-k} & \ddots & \ddots & \ddots & & \\ & \ddots & & & & \\ & & \ddots & & & \\ b.c. & & & & & \end{bmatrix}$$

This structure is called "block-Toeplitz with Toeplitz blocks" (BTTB) (when the b.c. respect it).
If the b.c. are periodic, we get "block-circulant with circulant blocks", BCCB.

This matrix is huge, we should try not to construct it explicitly!

Spectral decompositions

It turns out circulant and BCCB matrices have very special linear algebra properties.

Theorem

The columns of F^{-1} are eigenvectors of each circulant matrix:

$$\begin{bmatrix} p_0 & p_{n-1} & \cdots & p_1 \\ p_1 & p_0 & \cdots & p_2 \\ \vdots & \ddots & \ddots & \vdots \\ p_{n-1} & p_2 & \cdots & p_0 \end{bmatrix} \begin{bmatrix} z^0 \\ z^{n-i} \\ z^{n-2i} \\ \vdots \\ z^{-i} \end{bmatrix} = \lambda_i \begin{bmatrix} z^0 \\ z^{n-i} \\ z^{n-2i} \\ \vdots \\ z^{-i} \end{bmatrix}$$

with $\lambda_i = p_0 + p_1 z^i + p_2 z^{2i} + \cdots + p_{n-1} z^{(n-1)i}$: i.e., the eigenvalues are $DFT(\mathbf{p})$.

In other words, each circulant matrix $C(\mathbf{p})$ is $F^{-1}\,\mathrm{diag}(\mathrm{DFT}(\mathbf{p}))\,F$.

Working with circulant matrices

If $C(\mathbf{p}) = F^{-1}\,\mathrm{diag}(\mathrm{DFT}(\mathbf{p}))\,F$, then the solution of $C(\mathbf{p})\mathbf{x} = \mathbf{b}$ is

$$\mathbf{x} = C(\mathbf{p})^{-1}\mathbf{b} = F^{-1}\,\mathrm{diag}(\mathrm{DFT}(\mathbf{p}))^{-1}\,F\mathbf{b}$$

i.e., $\mathbf{x} = \mathrm{IDFT}(\mathrm{DFT}(\mathbf{b}) \oslash \mathrm{DFT}(\mathbf{p}))$. ($\oslash$ = elementwise division.)
or, more symmetrically,

$$\mathbf{x} = \mathrm{IDFT}(\mathrm{DFT}(\mathbf{b}) \oslash \mathrm{DFT}(\mathbf{p})).$$

Note that this formula does not depend on the choice of scaling in the various FFT implementations.

$O(n \log n)$ algorithm to invert circulant matrices.

Warning

Note how $\mathbf{p}$ has to be 'centered': for instance, the vector $\mathbf{p}$ corresponding to $b_i = p_1 x_{i-1} + p_0 x_i + p_{-1} x_{i+1}$ is $\mathbf{p} = [p_0, p_1, 0, 0, \dots, 0, p_{-1}]$.

The 2D case

Similarly, a BCCB matrix has eigenvector matrix $F^{-1} \otimes F^{-1}$ (i.e., DFT on columns, then DFT on rows of the image X).

Matlab has a `fft2` function (and a corresponding `ifft2`) to operate on matrices directly.

The BCCB matrix C generated by a ('centered') PSF P can be decomposed as

$$C = (F \otimes F)^{-1} \operatorname{diag}(\mathrm{DFT2(P)})(F \otimes F).$$

For an $m \times n$ image, C is $mn \times mn$, and its eigenvalues are the mn entries of $DFT2(P)$.

$$X = \mathrm{IDFT2}(\mathrm{DFT2}(B) \oslash \mathrm{DFT2}(P)).$$

Matlab example

```
X = imread('cameraman.tif'); X = double(X);

P = gaussianblur(5,11); %our code
B = conv2_centered(X, P, [5,11], 'cyclic'); %our code

P_padded = zeros(size(X));
P_padded(1:size(P,1), 1:size(P,2)) = P;
P_centered = circshift(P_padded, 1 - center);

X_reconstructed = ifft2(fft2(B) ./ fft2(P_centered));

subplot(1,3,1); imshow(uint8(X));
subplot(1,3,2); imshow(uint8(B));
subplot(1,3,3); imshow(uint8(X_reconstructed));
```

Noise

So far so good, but everything fails when noise is added. Even simply rounding to integer pixel intensities, `B = uin8(B)`.

We need regularization (as seen in Gianna's lecture).

$C = (F \otimes F)^{-1} \operatorname{diag}(\mathrm{DFT2}(\mathrm{P}))(F \otimes F)$, but what is its SVD instead?

Remember that F satisfies $F\overline{F}^T = nI$. Hence $\frac{1}{\sqrt{n}}F$ is 'conjugate-orthogonal' (unitary) which (long story short) is the correct thing to put in the SVD of a complex matrix.

The SVD of C

The SVD of C is

$$C = \underbrace{(F \otimes F)^{-1}}_{U} \underbrace{\mathrm{diag}(\mathrm{DFT2}(\mathrm{P}))\,\mathrm{diag}(\mathrm{D})}_{S} \underbrace{\mathrm{diag}(\mathrm{D})^{-1}(F \otimes F)}_{V^T},$$

where D is a matrix of complex numbers with $|d_{ii}| = 1$ chosen so that S is real positive (hence D is unitary, and $S = \mathrm{diag}(|\mathrm{DFT2}(\mathrm{P})|)$).

In practice, most of the time we can work with the decomposition $C = (F \otimes F)^{-1}\,\mathrm{diag}(\mathrm{DFT2}(\mathrm{P}))(F \otimes F)$ 'as if' it were an SVD, and alter the entries on its diagonal based on their magnitude.

Noise filtering

Truncated SVD: in the expression for C^{-1}, replace $\frac{1}{\sigma_i}$ with 0 if $|\sigma_i|$ is below a certain threshold.

Tikhonov / ridge regression: replace $\frac{1}{\sigma_i}$ with $\frac{\sigma_i}{\sigma_i^2+\tau^2}$, where τ is a suitably chosen parameter. I.e., replace a (complex) entry c of $DFT2(P)$ with $\frac{c}{|c|^2+\tau^2}$.

(Matlab example; worse results).

What if my BC are not periodic?

One can use similar transforms for reflective boundary conditions, which is arguably a lot better than periodic.

What about zero / arbitrary b.c.? Unfortunately, it is not true anymore that all BTTB matrices have the same basis of eigenvectors.

Trick: iterative methods (Arnoldi / CG / GMRES etc.).

- Allow one to solve large, sparse linear systems $Ax = b$ iteratively.
- Can exploit knowledge of how to solve linear systems with a 'similar' matrix $M \approx A$ to speed up the method.

Iterative methods

- CG (for $Ax = b$ with symmetric posdef A)
- GMRES (for $Ax = b$ with any square A)
- LSQR (for $\min \|Ax - b\|$ with possibly rectangular A, no Tikhonov)

Here we will experiment with GMRES.

```
>> gmres(A, b, [], tol, maxit, M)
```

GMRES converges 'faster' along components pertaining to signal, slower on noise.

$\implies$ doing just a few iterations of GMRES has a sort-of regularizing effect.

The problem with periodic boundary conditions / FFT provides a good preconditioner M for this matrix.

References

Hansen, Nagy, O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*
Short-ish book (ca. 120 pages) with this and much more (reflective boundary conditions, color images, cross-validation termination criteria. . .)