Sample 6-4

標本化

随伴作用素

画像処理特論

村松 正吾

動作確認: MATLAB R2020a

Sampling

Adjoint operator

Advanced Topics in Image Processing

Shogo MURAMATSU

Verified: MATLAB R2020a

準備

(Preparation)

```
close all
```

随伴作用素

(Adjoint operator)

以下のように内積を保存する作用素 T^* を作用素 T の随伴作用素と呼ぶ。(The operator T^* that preserves the inner product as follows is called an adjunct operator of operator T.)

$$\langle \{v[\mathbf{m}]\}_{\mathbf{m}}, T(\{u[\mathbf{n}]\}_{\mathbf{n}})\rangle = \langle T^*(\{v[\mathbf{m}]\}_{\mathbf{m}},), \{u[\mathbf{n}]\}_{\mathbf{n}}\rangle$$

この関係は行列の（エルミート）転置の一般化となっている。(This relationship is a generalization of the (Hermitian) transposition of a matrix.)

$$\langle \mathbf{v}, \mathbf{Au}\rangle = \langle \mathbf{Bv}, \mathbf{u}\rangle$$

```
% Generation of vectors
nDimV = 2;
nDimU = 2;
vecU = randn(nDimU,1) + 1j*randn(nDimU,1);
vecV = randn(nDimV,1) + 1j*randn(nDimV,1);

% Generation of a matrix
matA = randn(nDimU,nDimV);

% Inner product <v,Au>
innprodA = dot(vecV,matA*vecU);
```

```
% Inner product <A'v,u>
innprodB = dot(matA'*vecV,vecU);

% Absolute difference
err = abs(innprodA - innprodB);
disp(['|<v,Au> - <A''v,u>| = ' num2str(err)])
```

```
|<v,Au> - <A'v,u>| = 2.7756e-17
```

エルミート転置

(Herimitian transpose)

エルミート転置は複素共役転置を意味する。(The Hermitian transposition implies a complex conjugate transposition.)

```
% Complex conjugate transposition of matrix A
matB = ctranspose(matA);

% Absolute difference
err = norm(matA' - matB,'fro');
disp(['||B - A''||F = ' num2str(err)])
```

```
||B - A'||F = 0
```

二変量間引き処理の随伴作用素

(Adjoint operator of bivariate downsampling)

```
% Input array size
N1 =6;
N2 =4;
% Downsampling factor
M1 =2;
M2 =2;

% Definition of bivariate separable downsampling
downsample2 = @(x,n) ...
    shiftdim(downsample(...
    shiftdim(downsample(x,...
    n(1)),1),...
    n(2)),1);

% Find the matrix representation of the bivariate downsampling
N   = N1*N2;
T = [];
for idx = 1:N
    % Generating a standard basis vector
    e = zeros(N1,N2);
    e(idx) = 1;
    % Response to the standard basis vector
    t = downsample2(e,[M1 M2]);
    T(:,idx) = t(:);
end
```

行列表現 (Matrix represntation)

```
% Matrix representation of the bivariate downsampling
T
```

```
T = 6×24
     1     0     0     0     0     0     0     0     0     0     0     0     0 ···
     0     0     1     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
% Adjoint matrix of the bivariate downsampling
T'
```

```
ans = 24×6
     1     0     0     0     0     0
     0     0     0     0     0     0
     0     1     0     0     0     0
     0     0     0     0     0     0
     0     0     1     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     ⋮
```

随伴作用素 (Adjoint operator)

$$T^{*}(\{v[\mathbf{m}]\}_{\mathbf{m}}) = \mathrm{vec}_{\Omega_{\mathrm{u}}}^{-1} \circ \mathbf{T}^{H}\mathrm{vec}_{\Omega_{\mathrm{v}}}(\{v[\mathbf{m}]\}_{\mathbf{m}})$$

```
% Adjoint operator T*
adjOp = @(x) reshape(T'*x(:),[N1 N2]);

% Generation of an input array u
arrayU = randn(N1,N2);
```

$$\{v[\mathbf{m}]\}_{\mathbf{m}} = T(\{u[\mathbf{n}]\}_{\mathbf{n}})$$

```
% Downsampling (v=Tu)
arrayV = downsample2(arrayU,[M1 M2]);

% Array generation in the same domain with arrayV
arrayY = randn(size(arrayV),'like',arrayV)
```

```
arrayY = 3×2
    1.0391    0.6601
   -1.1176   -0.0679
    1.2607   -0.1952
```

$$\langle \mathbf{y}, \mathbf{v} \rangle = \langle \mathbf{y}, \mathbf{Tu} \rangle$$

```
% Inner product <y,v>=<y,Tu>
```

```
innprodA = dot(arrayY(:),arrayV(:));
```

間引き処理の随伴作用素は零値挿入処理 (The adjoint operator of downsampling is upsampling.)

$\mathbf{r} = \mathbf{T}^H\mathbf{v}$

```
% Adjoint operation of downsampling (r=T'v)
arrayR = adjOp(arrayY)
```

```
arrayR = 6×4
    1.0391         0    0.6601         0
         0         0         0         0
   -1.1176         0   -0.0679         0
         0         0         0         0
    1.2607         0   -0.1952         0
         0         0         0         0
```

$\langle \mathbf{r}, \mathbf{u} \rangle = \langle \mathbf{T}^H\mathbf{y}, \mathbf{u} \rangle$

```
% Inner product <r,u>=<T'v,u>
innprodB = dot(arrayR(:),arrayU(:));

% Verify the preservation of the inner product
err = abs(innprodA - innprodB);
disp(['|<y,Tu> - <T''y,u>| = ' num2str(err)])
```

```
|<y,Tu> - <T'y,u>| = 4.4409e-16
```