

6.5. Let $P \in \mathbb{C}^{m \times m}$ be a nonzero projector. Show that $\|P\|_2 \geq 1$, with equality if and only if P is an orthogonal projector.

1) Let $u \in \mathbb{C}^m \ni Pu = v$. WLOG, let $\|u\|_2 = 1$ and $\|v\|_2 = 1$.

Thus,

$$1 = \|u\|_2 = \|Pu\|_2 \leq \|P\|_2 \|v\|_2 = \|P\|_2.$$

If P is orthogonal this inequality does not change.
It follows,

$$\sup_{\|x\|_2=1} \|Px\|_2 \geq \|Pu\|_2 = \|u\|_2 = 1.$$

Assuming equality,

$$\sup_{\|x\|_2=1} \|x\|_2 = 1 = \|P\|_2 = \sup_{\|x\|_2=1} \|Px\|_2$$

Thus, $\|x\|_2 = \|Px\|_2$ which shows P is orthogonal.

2. [Q2] (15 points) Review our analysis of the bound on the relative forward error of singular value computation by using a backward stable eigenvalue algorithm for $A^T A$. That is, $\frac{|\tilde{\sigma}_k - \sigma_k|}{\sigma_k} \leq \mathcal{O}\left(\frac{\sigma_1^2}{\sigma_k^2} \epsilon_{mach}\right)$, where $\tilde{\sigma}_k = \sqrt{\tilde{\lambda}_k}$ and $\tilde{\lambda}_k$ denotes the computed k -th largest eigenvalue of $A^T A$. Instead, if we use a backward stable eigenvalue algorithm for $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$, show that the relative forward error of singular value computation would be bounded by $\mathcal{O}\left(\frac{\sigma_1}{\sigma_k} \epsilon_{mach}\right)$, assuming that square root computation is exact. Explain the advantage of the new error bound.

2) Let $H = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$. It follows,

$$H \begin{bmatrix} V & V \\ U & -U \end{bmatrix} = \begin{bmatrix} A^T U & -A^T U \\ AV & AV \end{bmatrix} = \begin{bmatrix} V\Sigma & -V\Sigma \\ U\Sigma & U\Sigma \end{bmatrix} = \begin{bmatrix} V & V \\ U & -U \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix}$$

Thus $\lambda(H) = \pm \sigma_i(A)$.

By Bauer-Fike, $|\hat{\lambda}_k - \lambda_k(H)| = |\hat{\lambda}_k(H + \Delta H) - \lambda_k(H)| \leq \|\Delta H\|_2 = O(\Sigma_m) \|H\|_2 = O(\Sigma_m \lambda_1) = O(\Sigma_m \sigma_1)$

$$|\hat{\lambda}_k - \lambda_k(H)| = |\hat{\sigma}_k - \sigma_k| \leq O(\Sigma_n \sigma_1) \Rightarrow \frac{|\hat{\sigma}_k - \sigma_k|}{\sigma_k} \leq O\left(\Sigma_n \frac{\sigma_1}{\sigma_k}\right)$$

This bound is useful if $\sigma_k \ll \sigma_1$ since $\frac{\sigma_1}{\sigma_k}$ is a tighter bound than $\frac{\sigma_1^2}{\sigma_k^2}$.

3. [Q3] (15 points) Read the introduction to the Golub-Kahan-Lanczos method, at <http://www.netlib.org/utk/people/JackDongarra/etemplates/node198.html> and the uploaded code implementation `HW6_GKLsvds.m`.

- (a) Give a general description of the functionality of GKL; describe the main difference between the

3. [Q3] (15 points) Read the introduction to the Golub-Kahan-Lanczos method, at <http://www.netlib.org/utk/people/JackDongarra/etemplates/node198.html> and the uploaded code implementation `HW6_GKLSvds.m`.
 (a) Give a general description of the functionality of GKL; describe the main difference between the original GKL algorithm and the MATLAB code.
 (b) Download the zipped file `HW6_pics.zip`, unzip it, load the first jpeg file by

```
picA = double(imread('picA.jpg'));
```

and run

```
rk = 160;
tic; [Us1,Ss1,Vs1] = HW6_GKLSvds(picA(:,:,1),rk); toc;
tic; [Us2,Ss2,Vs2] = HW6_GKLSvds(picA(:,:,2),rk); toc;
tic; [Us3,Ss3,Vs3] = HW6_GKLSvds(picA(:,:,3),rk); toc;
tic; [U1,S1,V1] = svd(picA(:,:,1),0); toc;
tic; [U2,S2,V2] = svd(picA(:,:,2),0); toc;
tic; [U3,S3,V3] = svd(picA(:,:,3),0); toc;
```

Then, run MATLAB's `whos` to see the memory used by `picA`, and by `Us1, Vs1, Us2, Vs2, Us3` and `Vs3` all together. Compare the timing used for computing and the memory used for storing the full and partial SVD of this picture.

(Note: we are competing MATLAB code with the built-in C/FORTRAN code in timing, and our timing should improve considerably if our GKL code is in C/FORTRAN)

- (c) Finally, run MATLAB's command

```
picAh = zeros(size(picA));
picAh(:,:,1) = Us1*Ss1*Vs1';
picAh(:,:,2) = Us2*Ss2*Vs2';
picAh(:,:,3) = Us3*Ss3*Vs3';

disp([norm(picAh(:,:,1)-picA(:,:,1),'fro')/norm(picA(:,:,1),'fro') ...
norm(picAh(:,:,2)-picA(:,:,2),'fro')/norm(picA(:,:,2),'fro') ...
norm(picAh(:,:,3)-picA(:,:,3),'fro')/norm(picA(:,:,3),'fro')]);

figure(1); imshow(uint8(picA)); axis equal;
figure(2); imshow(uint8(picAh)); axis equal;
```

Are you satisfied with the quality of the image generated by `picAh`? If not, let `rk = 320`, rerun `HW6_GKLSvds`, compare the timing and memory cost for computing the partial SVD. Then show the images again. Are you satisfied now?

Repeat the above procedure for the other three pictures. Make some general comments on the computation and use of partial SVD for compressing images. In particular, give an estimate of the arithmetic cost of this partial SVD and full SVD applied to an image of dimension m -by- n , in a form of $\mathcal{O}(\cdot)$. For a given rank $r \ll \min\{m, n\}$, how does the cost of partial SVD compared to full SVD as $\min\{m, n\}$ increases? In a recent development by random sketching, full orthogonalization of r vectors of elements m or n ($r \ll \min\{m, n\}$) needs only $\mathcal{O}(r^3) + \mathcal{O}(\max\{m, n\}r)$ flops. Compare the cost of partial and full SVD if such a fast orthogonalization can be used.

- (d) (4 extra points for fun, only for those who finished (a)(b)(c)). Search the title of each artwork, the name of the artist, the approximate year of creation, and the current location of the artwork. Info of 4 paintings qualifies full extra credit. *Do your own research, instead of using others' findings, even for this leisure problem.*

a) GKL uses a double recursion relation to find the bidiagonalization. The MATLAB Code finds the first k singular values and corresponding vectors based off this bidiagonalization!

b) Timing Full is ~5-6 times longer than partial.
 Memory Full is ~20 times larger

c) Timing than partial.

Memory Full is ~20 times larger than partial.

c) Timing Full is ~2-3 times longer than partial.

Memory Full is ~6-7 times larger than partial.

The pictures have more quality.

Partial SVD saves a lot of memory and is faster, but loses quality.

WLOG Assume Mn .

Partial rank = $k \sim O(mn)$

Full $\sim O(mn^2)$

The cost increases linearly for a partial SVD because rank would be a fixed value.

The full SVD increases quadratically because of the $O(r^3)$

Partial

If $r \ll n$ then $O(r^3) + O(mr) < O(mnr)$

Full

$O(n^3) + O(mn) < O(mn^2)$ if $n \ll m$

d) A: Last Judgement

Michelangelo

1536-1541

Sistine Chapel Vatican

B: Last Supper

Leonardo Da Vinci

1495-1498

Convent of Santa Maria delle Grazie Milan

C: Renoir Le Moulin De La Galette

Pierre-Auguste Renoir

1876

Musée d'Orsay Paris

D: Night Watch

Rembrandt van Rijn

D: Night Watch
Rembrandt van Rijn
1642
Rijksmuseum Amsterdam

E: Girl with a Pearl Earring
Johannes Vermeer
1665
Mauritshuis Museum The Hague Netherlands

F: Potato Eaters
Vincent Van Gogh
1885
Van Gogh Museum Amsterdam

G: The Starry Night
Vincent Van Gogh
1889
Museum of Modern Art New York City

```
disp("PicA")
picA = double(imread('picA.jpg'));
[pic, pich] = comp(picA,160);
figure(1); imshow(uint8(pic)); axis equal;
figure(2); imshow(uint8(pich)); axis equal;

[~, pich] = comp(picA,320);
figure(3); imshow(uint8(pich)); axis equal;

disp("PicB")
picB = double(imread('picB.jpg'));
[pic, pich] = comp(picB,160);
figure(4); imshow(uint8(pic)); axis equal;
figure(5); imshow(uint8(pich)); axis equal;

[~, pich] = comp(picB,320);
figure(6); imshow(uint8(pich)); axis equal;

disp("PicC")
picC = double(imread('picC.jpg'));
[pic, pich] = comp(picC,160);
figure(7); imshow(uint8(pic)); axis equal;
figure(8); imshow(uint8(pich)); axis equal;

[~, pich] = comp(picC,320);
figure(9); imshow(uint8(pich)); axis equal;

disp("PicD")
picD = double(imread('picD.jpg'));
[pic, pich] = comp(picD,160);
figure(10); imshow(uint8(pic)); axis equal;
figure(11); imshow(uint8(pich)); axis equal;

[~, pich] = comp(picD,320);
figure(12); imshow(uint8(pich)); axis equal;

disp("PicE")
picE = double(imread('picE.jpg'));
[pic, pich] = comp(picE,160);
figure(13); imshow(uint8(pic)); axis equal;
figure(14); imshow(uint8(pich)); axis equal;

[~, pich] = comp(picE,320);
figure(15); imshow(uint8(pich)); axis equal;

disp("PicF")
picF = double(imread('picF.jpg'));
[pic, pich] = comp(picF,160);
figure(16); imshow(uint8(pic)); axis equal;
figure(17); imshow(uint8(pich)); axis equal;

[~, pich] = comp(picF,320);
```

```

figure(18); imshow(uint8(pich)); axis equal;

disp("PicG")
picG = double(imread('picG.jpg'));
[pic, pich] = comp(picG,160);
figure(19); imshow(uint8(pic)); axis equal;
figure(20); imshow(uint8(pich)); axis equal;

[~, pich] = comp(picG,320);
figure(21); imshow(uint8(pich)); axis equal;

function [pic, pich] = comp(pic,rk)
tic; [Us1,Ss1,Vs1] = HW6_GKLsvds(pic(:,:,:,1),rk); toc;
tic; [Us2,Ss2,Vs2] = HW6_GKLsvds(pic(:,:,:,2),rk); toc;
tic; [Us3,Ss3,Vs3] = HW6_GKLsvds(pic(:,:,:,3),rk); toc;
tic; [U1,S1,V1] = svd(pic(:,:,:,1),0); toc;
tic; [U2,S2,V2] = svd(pic(:,:,:,2),0); toc;
tic; [U3,S3,V3] = svd(pic(:,:,:,3),0); toc;

whos pic
whos Us1 Vs1 Us2 Vs2 Us3 Vs3
whos U1 V1 U2 V2 U3 V3

pich = zeros(size(pic));
pich(:,:,:1) = Us1*Ss1*Vs1';
pich(:,:,:2) = Us2*Ss2*Vs2';
pich(:,:,:3) = Us3*Ss3*Vs3';

disp([norm(pich(:,:,:,1)-pic(:,:,:,1),'fro')/norm(pic(:,:,:,1),'fro') ...
norm(pich(:,:,:,2)-pic(:,:,:,2),'fro')/norm(pic(:,:,:,2),'fro') ...
norm(pich(:,:,:,3)-pic(:,:,:,3),'fro')/norm(pic(:,:,:,3),'fro'))]);
end

function [U,S,V] = HW6_GKLsvds(A,k)

% The Golub-Kahan-Lanczos bidiagonalization
%
% Input:
% A      The matrix for which we are computing largest singular values
%        should be large to show the competitiveness of iterative method
% k      The number of singular values wanted
%
% Output:
% S      The k by k diagonal matrix of approximate dominant singular values
% U,V   The k approximate left and right singular vectors
%
% Copyright (c) F. Xue 10/21/2017

[m,n] = size(A);
v_k = randn(n,1);
v_k = v_k/norm(v_k);
maxiter = min([m n max([ceil(1.2*k) k+5])]);
beta_km1 = 0;
u_km1 = zeros(m,1);

```

```

alpha_all = zeros(maxiter,1);
beta_all = zeros(maxiter+1,1);
U = zeros(m,maxiter);
V = zeros(n,maxiter+1);
V(:,1) = v_k;

for iter = 1 : maxiter
    u_k = A*v_k-beta_km1*u_km1;

    for jj = 1 : iter-1
        u_k = u_k - U(:,jj)*(U(:,jj)'*u_k);
    end
    %u_k = u_k - U(:,1:iter-1)*(U(:,1:iter-1)'*u_k);

    alpha_k = norm(u_k);
    u_k = u_k/alpha_k;
    v_kp1 = (u_k'*A)'-alpha_k*v_k;

    for jj = 1 : iter
        v_kp1 = v_kp1 - V(:,jj)*(V(:,jj)']*v_kp1);
    end
    %v_kp1 = v_kp1 - V(:,1:iter)*(V(:,1:iter)']*v_kp1);

    beta_k = norm(v_kp1);
    v_kp1 = v_kp1/beta_k;

    alpha_all(iter) = alpha_k;
    beta_all(iter+1) = beta_k;

    U(:,iter) = u_k;
    V(:,iter+1) = v_kp1;

    u_km1 = u_k;
    v_k = v_kp1;
    beta_km1 = beta_k;
end

B = spdiags([alpha_all beta_all(1:end-1)],0:1,maxiter,maxiter);
[Us,S,Vs] = svd(full(B));
U = U*Us(:,1:k);
V = V(:,1:maxiter)*Vs(:,1:k);
S = S(1:k,1:k);

end

PicA
Elapsed time is 0.770402 seconds.
Elapsed time is 0.673358 seconds.
Elapsed time is 0.785784 seconds.
Elapsed time is 3.971310 seconds.
Elapsed time is 6.780229 seconds.
Elapsed time is 6.408222 seconds.

      Name          Size            Bytes   Class         Attributes

```

<i>pic</i>	2304x2092x3	115679232	double	
<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>Us1</i>	2304x160	2949120	double	
<i>Us2</i>	2304x160	2949120	double	
<i>Us3</i>	2304x160	2949120	double	
<i>Vs1</i>	2092x160	2677760	double	
<i>Vs2</i>	2092x160	2677760	double	
<i>Vs3</i>	2092x160	2677760	double	
<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>U1</i>	2304x2092	38559744	double	
<i>U2</i>	2304x2092	38559744	double	
<i>U3</i>	2304x2092	38559744	double	
<i>V1</i>	2092x2092	35011712	double	
<i>V2</i>	2092x2092	35011712	double	
<i>V3</i>	2092x2092	35011712	double	
0.0978	0.0954	0.0968		
<i>Elapsed time is 3.327442 seconds.</i>				
<i>Elapsed time is 2.975858 seconds.</i>				
<i>Elapsed time is 3.229897 seconds.</i>				
<i>Elapsed time is 6.584234 seconds.</i>				
<i>Elapsed time is 6.210967 seconds.</i>				
<i>Elapsed time is 5.864038 seconds.</i>				
<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>pic</i>	2304x2092x3	115679232	double	
<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>Us1</i>	2304x320	5898240	double	
<i>Us2</i>	2304x320	5898240	double	
<i>Us3</i>	2304x320	5898240	double	
<i>Vs1</i>	2092x320	5355520	double	
<i>Vs2</i>	2092x320	5355520	double	
<i>Vs3</i>	2092x320	5355520	double	
<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>U1</i>	2304x2092	38559744	double	
<i>U2</i>	2304x2092	38559744	double	
<i>U3</i>	2304x2092	38559744	double	
<i>V1</i>	2092x2092	35011712	double	
<i>V2</i>	2092x2092	35011712	double	
<i>V3</i>	2092x2092	35011712	double	
0.0675	0.0674	0.0675		

PicB
Elapsed time is 2.853764 seconds.

```
Elapsed time is 2.543524 seconds.  
Elapsed time is 2.052792 seconds.  
Elapsed time is 6.670744 seconds.  
Elapsed time is 6.595645 seconds.  
Elapsed time is 6.968629 seconds.
```

Name	Size	Bytes	Class	Attributes
pic	2304x4089x3	226105344	double	
Name	Size	Bytes	Class	Attributes
Us1	2304x160	2949120	double	
Us2	2304x160	2949120	double	
Us3	2304x160	2949120	double	
Vs1	4089x160	5233920	double	
Vs2	4089x160	5233920	double	
Vs3	4089x160	5233920	double	
Name	Size	Bytes	Class	Attributes
U1	2304x2304	42467328	double	
U2	2304x2304	42467328	double	
U3	2304x2304	42467328	double	
V1	4089x4089	133759368	double	
V2	4089x4089	133759368	double	
V3	4089x4089	133759368	double	

0.0997 0.1122 0.1243

```
Elapsed time is 3.269452 seconds.  
Elapsed time is 3.589012 seconds.  
Elapsed time is 3.125423 seconds.  
Elapsed time is 6.625222 seconds.  
Elapsed time is 7.603165 seconds.  
Elapsed time is 7.613257 seconds.
```

Name	Size	Bytes	Class	Attributes
pic	2304x4089x3	226105344	double	
Name	Size	Bytes	Class	Attributes
Us1	2304x320	5898240	double	
Us2	2304x320	5898240	double	
Us3	2304x320	5898240	double	
Vs1	4089x320	10467840	double	
Vs2	4089x320	10467840	double	
Vs3	4089x320	10467840	double	
Name	Size	Bytes	Class	Attributes
U1	2304x2304	42467328	double	
U2	2304x2304	42467328	double	
U3	2304x2304	42467328	double	
V1	4089x4089	133759368	double	

V2	4089x4089	133759368	double
V3	4089x4089	133759368	double

0.0878 0.0994 0.1103

PicC

Elapsed time is 1.047781 seconds.
Elapsed time is 1.073389 seconds.
Elapsed time is 1.636405 seconds.
Elapsed time is 5.728921 seconds.
Elapsed time is 5.652349 seconds.
Elapsed time is 8.863288 seconds.

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

pic	2304x3100x3	171417600	double	
-----	-------------	-----------	--------	--

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

Us1	2304x160	2949120	double	
Us2	2304x160	2949120	double	
Us3	2304x160	2949120	double	
Vs1	3100x160	3968000	double	
Vs2	3100x160	3968000	double	
Vs3	3100x160	3968000	double	

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

U1	2304x2304	42467328	double	
U2	2304x2304	42467328	double	
U3	2304x2304	42467328	double	
V1	3100x3100	76880000	double	
V2	3100x3100	76880000	double	
V3	3100x3100	76880000	double	

0.1148 0.1005 0.1162

Elapsed time is 2.713616 seconds.
Elapsed time is 2.598876 seconds.
Elapsed time is 2.188496 seconds.
Elapsed time is 9.426360 seconds.
Elapsed time is 9.278503 seconds.
Elapsed time is 8.432735 seconds.

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

pic	2304x3100x3	171417600	double	
-----	-------------	-----------	--------	--

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

Us1	2304x320	5898240	double	
Us2	2304x320	5898240	double	
Us3	2304x320	5898240	double	
Vs1	3100x320	7936000	double	
Vs2	3100x320	7936000	double	
Vs3	3100x320	7936000	double	

Name	Size	Bytes	Class	Attributes
U1	2304x2304	42467328	double	
U2	2304x2304	42467328	double	
U3	2304x2304	42467328	double	
V1	3100x3100	76880000	double	
V2	3100x3100	76880000	double	
V3	3100x3100	76880000	double	
		0.0842	0.0758	0.0897

PicD

Elapsed time is 1.283196 seconds.
Elapsed time is 1.496474 seconds.
Elapsed time is 1.514272 seconds.
Elapsed time is 4.979288 seconds.
Elapsed time is 7.649908 seconds.
Elapsed time is 6.803081 seconds.

Name	Size	Bytes	Class	Attributes
pic	2304x2750x3	152064000	double	

Name	Size	Bytes	Class	Attributes
Us1	2304x160	2949120	double	
Us2	2304x160	2949120	double	
Us3	2304x160	2949120	double	
Vs1	2750x160	3520000	double	
Vs2	2750x160	3520000	double	
Vs3	2750x160	3520000	double	

Name	Size	Bytes	Class	Attributes
U1	2304x2304	42467328	double	
U2	2304x2304	42467328	double	
U3	2304x2304	42467328	double	
V1	2750x2750	60500000	double	
V2	2750x2750	60500000	double	
V3	2750x2750	60500000	double	

0.0706 0.0809 0.1150

Elapsed time is 3.174940 seconds.
Elapsed time is 2.167861 seconds.
Elapsed time is 1.709499 seconds.
Elapsed time is 7.843681 seconds.
Elapsed time is 7.523695 seconds.
Elapsed time is 8.146388 seconds.

Name	Size	Bytes	Class	Attributes
pic	2304x2750x3	152064000	double	

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

<i>Us1</i>	2304x320	5898240	<i>double</i>
<i>Us2</i>	2304x320	5898240	<i>double</i>
<i>Us3</i>	2304x320	5898240	<i>double</i>
<i>Vs1</i>	2750x320	7040000	<i>double</i>
<i>Vs2</i>	2750x320	7040000	<i>double</i>
<i>Vs3</i>	2750x320	7040000	<i>double</i>

Name	Size	Bytes	Class	Attributes
<i>U1</i>	2304x2304	42467328	<i>double</i>	
<i>U2</i>	2304x2304	42467328	<i>double</i>	
<i>U3</i>	2304x2304	42467328	<i>double</i>	
<i>V1</i>	2750x2750	60500000	<i>double</i>	
<i>V2</i>	2750x2750	60500000	<i>double</i>	
<i>V3</i>	2750x2750	60500000	<i>double</i>	

0.0468 0.0563 0.0851

PicE

Elapsed time is 1.138180 seconds.
Elapsed time is 1.147484 seconds.
Elapsed time is 1.044296 seconds.
Elapsed time is 4.660685 seconds.
Elapsed time is 5.227930 seconds.
Elapsed time is 4.657759 seconds.

Name	Size	Bytes	Class	Attributes
<i>pic</i>	2304x1968x3	108822528	<i>double</i>	

Name	Size	Bytes	Class	Attributes
<i>Us1</i>	2304x160	2949120	<i>double</i>	
<i>Us2</i>	2304x160	2949120	<i>double</i>	
<i>Us3</i>	2304x160	2949120	<i>double</i>	
<i>Vs1</i>	1968x160	2519040	<i>double</i>	
<i>Vs2</i>	1968x160	2519040	<i>double</i>	
<i>Vs3</i>	1968x160	2519040	<i>double</i>	

Name	Size	Bytes	Class	Attributes
<i>U1</i>	2304x1968	36274176	<i>double</i>	
<i>U2</i>	2304x1968	36274176	<i>double</i>	
<i>U3</i>	2304x1968	36274176	<i>double</i>	
<i>V1</i>	1968x1968	30984192	<i>double</i>	
<i>V2</i>	1968x1968	30984192	<i>double</i>	
<i>V3</i>	1968x1968	30984192	<i>double</i>	

0.0884 0.1000 0.1110

Elapsed time is 3.031735 seconds.
Elapsed time is 2.665102 seconds.
Elapsed time is 2.973368 seconds.
Elapsed time is 4.451723 seconds.

Elapsed time is 4.604524 seconds.

Elapsed time is 4.576955 seconds.

Name	Size	Bytes	Class	Attributes
pic	2304x1968x3	108822528	double	
<hr/>				
Name	Size	Bytes	Class	Attributes
Us1	2304x320	5898240	double	
Us2	2304x320	5898240	double	
Us3	2304x320	5898240	double	
Vs1	1968x320	5038080	double	
Vs2	1968x320	5038080	double	
Vs3	1968x320	5038080	double	
Name	Size	Bytes	Class	Attributes
U1	2304x1968	36274176	double	
U2	2304x1968	36274176	double	
U3	2304x1968	36274176	double	
V1	1968x1968	30984192	double	
V2	1968x1968	30984192	double	
V3	1968x1968	30984192	double	
0.0706	0.0797	0.0883		

PicF

Elapsed time is 1.578696 seconds.

Elapsed time is 1.965002 seconds.

Elapsed time is 1.586833 seconds.

Elapsed time is 9.179669 seconds.

Elapsed time is 10.074482 seconds.

Elapsed time is 9.680614 seconds.

Name	Size	Bytes	Class	Attributes
pic	2304x3243x3	179324928	double	
<hr/>				
Name	Size	Bytes	Class	Attributes
Us1	2304x160	2949120	double	
Us2	2304x160	2949120	double	
Us3	2304x160	2949120	double	
Vs1	3243x160	4151040	double	
Vs2	3243x160	4151040	double	
Vs3	3243x160	4151040	double	
Name	Size	Bytes	Class	Attributes
U1	2304x2304	42467328	double	
U2	2304x2304	42467328	double	
U3	2304x2304	42467328	double	
V1	3243x3243	84136392	double	
V2	3243x3243	84136392	double	
V3	3243x3243	84136392	double	

0.2017 0.1944 0.2872

Elapsed time is 4.169961 seconds.
Elapsed time is 3.205528 seconds.
Elapsed time is 3.356740 seconds.
Elapsed time is 10.071966 seconds.
Elapsed time is 9.545252 seconds.
Elapsed time is 9.061124 seconds.

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

pic	2304x3243x3	179324928	double	
-----	-------------	-----------	--------	--

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

Us1	2304x320	5898240	double	
Us2	2304x320	5898240	double	
Us3	2304x320	5898240	double	
Vs1	3243x320	8302080	double	
Vs2	3243x320	8302080	double	
Vs3	3243x320	8302080	double	

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

U1	2304x2304	42467328	double	
U2	2304x2304	42467328	double	
U3	2304x2304	42467328	double	
V1	3243x3243	84136392	double	
V2	3243x3243	84136392	double	
V3	3243x3243	84136392	double	

0.1628 0.1588 0.2384

PicG
Elapsed time is 1.544835 seconds.
Elapsed time is 1.314507 seconds.
Elapsed time is 1.540077 seconds.
Elapsed time is 6.722772 seconds.
Elapsed time is 8.071382 seconds.
Elapsed time is 8.192187 seconds.

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

pic	2304x2910x3	160911360	double	
-----	-------------	-----------	--------	--

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

Us1	2304x160	2949120	double	
Us2	2304x160	2949120	double	
Us3	2304x160	2949120	double	
Vs1	2910x160	3724800	double	
Vs2	2910x160	3724800	double	
Vs3	2910x160	3724800	double	

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

<i>U1</i>	2304x2304	42467328	<i>double</i>
<i>U2</i>	2304x2304	42467328	<i>double</i>
<i>U3</i>	2304x2304	42467328	<i>double</i>
<i>V1</i>	2910x2910	67744800	<i>double</i>
<i>V2</i>	2910x2910	67744800	<i>double</i>
<i>V3</i>	2910x2910	67744800	<i>double</i>

0.2010 0.1677 0.1509

Elapsed time is 3.447620 seconds.
Elapsed time is 3.317635 seconds.
Elapsed time is 3.315637 seconds.
Elapsed time is 8.161908 seconds.
Elapsed time is 8.293347 seconds.
Elapsed time is 8.263234 seconds.

<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>pic</i>	2304x2910x3	160911360	<i>double</i>	

<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>Us1</i>	2304x320	5898240	<i>double</i>	
<i>Us2</i>	2304x320	5898240	<i>double</i>	
<i>Us3</i>	2304x320	5898240	<i>double</i>	
<i>Vs1</i>	2910x320	7449600	<i>double</i>	
<i>Vs2</i>	2910x320	7449600	<i>double</i>	
<i>Vs3</i>	2910x320	7449600	<i>double</i>	

<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>U1</i>	2304x2304	42467328	<i>double</i>	
<i>U2</i>	2304x2304	42467328	<i>double</i>	
<i>U3</i>	2304x2304	42467328	<i>double</i>	
<i>V1</i>	2910x2910	67744800	<i>double</i>	
<i>V2</i>	2910x2910	67744800	<i>double</i>	
<i>V3</i>	2910x2910	67744800	<i>double</i>	

0.1629 0.1358 0.1232







































Published with MATLAB® R2023b