

---

# Table of Contents

.....	1
1 .....	1
2 .....	5
3 .....	5
4 .....	11
5 .....	12

ECE 418 Digital Video Problems: Lattices DongKyu Kim

```
clear all; close all; clc;
```

1

```
% Lattice and Reciprocal Lattice V, V^
V1 = [1,1;-1,1];
V2 = [1,-1/2;0,sqrt(3)/2];
V1r = V1^(-1).';
V2r = V2^(-1).';
n = -10:10;
n_vec = [kron(ones(1,21),n)',kron(n,ones(1,21))']';

r1 = V1*n_vec;
r2 = V2*n_vec;
rr1 = V1r*n_vec;
rr2 = V2r*n_vec;

% Lattice and Reciprocal Lattice U, U^
M = [2,0;1,2];
U1 = V1*M;
U2 = V2*M;
U1r = U1^(-1).';
U2r = U2^(-1).';

ru1 = U1*n_vec;
ru2 = U2*n_vec;
rur1 = U1r*n_vec;
rur2 = U2r*n_vec;

% Non overlapping Points
star1 = setdiff(r1.',ru1.', 'rows').';
star2 = setdiff(r2.',ru2.', 'rows').';
starr1 = setdiff(rur1.',rur1.', 'rows').'; %Lvhat < Luhut rur1 is denser
starr2 = setdiff(rur2.',rur2.', 'rows').';
% I think due to precision error matlab misses some points for V2

%Plotting
% I was being lazy and plotted the whole V, U first then plotted *
points.
figure(1);
```

---

```

hold on;
plot(r1(1,:),r1(2,:), 'x');
plot(rul(1,:),rul(2,:), 'o');
plot(star1(1,:),star1(2,:), '*');
legend('V', 'U', 'V-U');
hold off;
title('points in the lattice of 1st V with U');

figure(2);
hold on;
plot(rr1(1,:),rr1(2,:), 'x');
plot(rur1(1,:),rur1(2,:), 'o');
plot(starr1(1,:),starr1(2,:), '*');
legend('V', 'U', 'U-V');
hold off;
title('points in the reciprocal lattice of 1st V with U');

figure(3);
hold on;
plot(r2(1,:),r2(2,:), 'x');
plot(ru2(1,:),ru2(2,:), 'o');
plot(star2(1,:),star2(2,:), '*');
legend('V', 'U', 'V-U');
hold off;
title('points in the lattice of 2nd V with U');

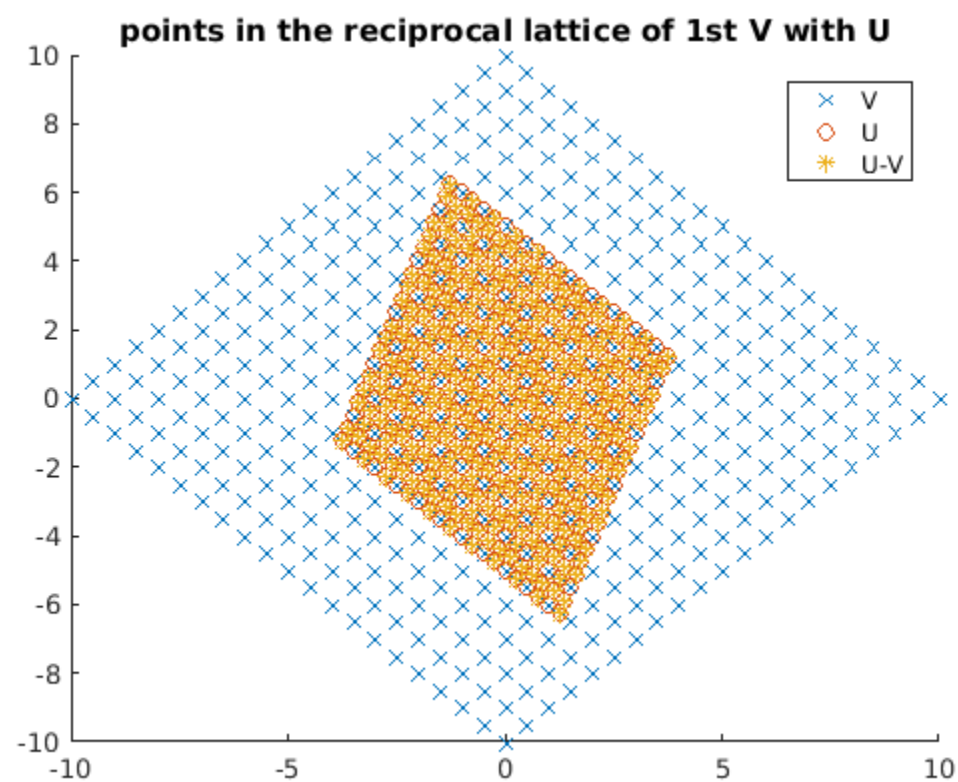
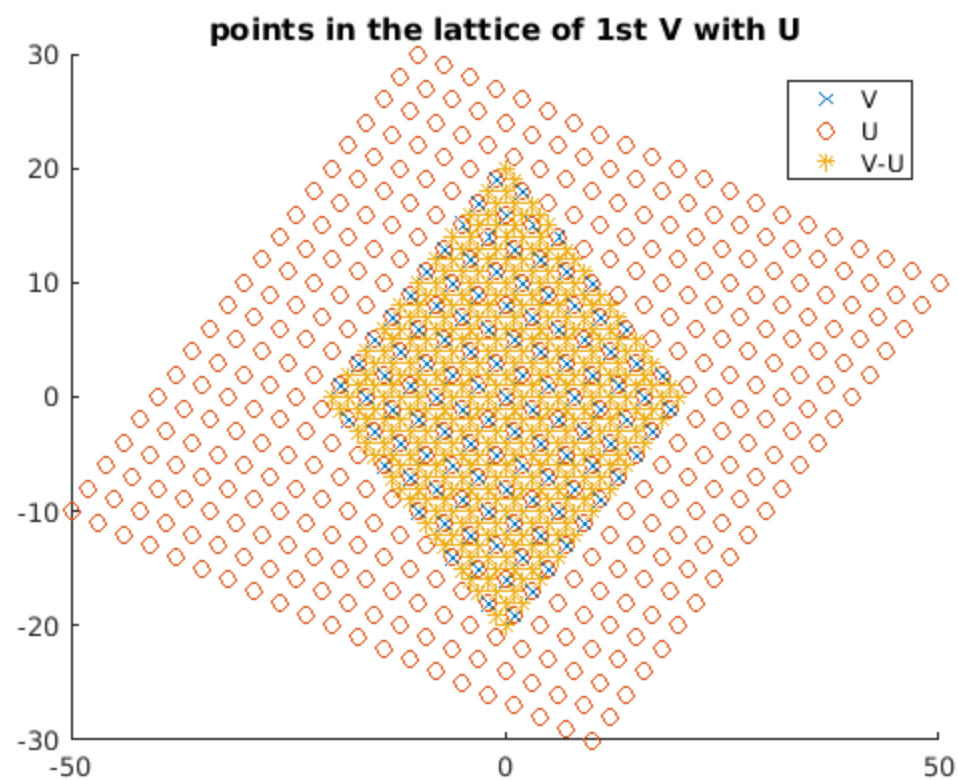
figure(5);
hold on;
plot(rr2(1,:),rr2(2,:), 'x');
plot(rur2(1,:),rur2(2,:), 'o');
plot(starr2(1,:),starr2(2,:), '*');
legend('V', 'U', 'U-V');
hold off;
title('points in the reciprocal lattice of 2nd V with U');

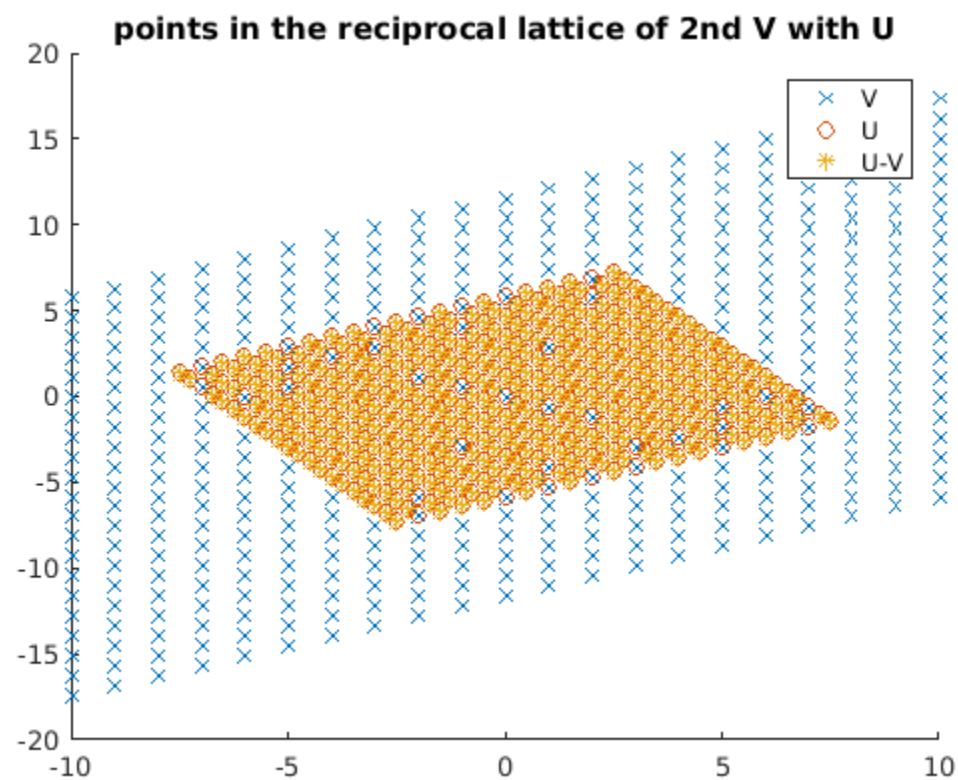
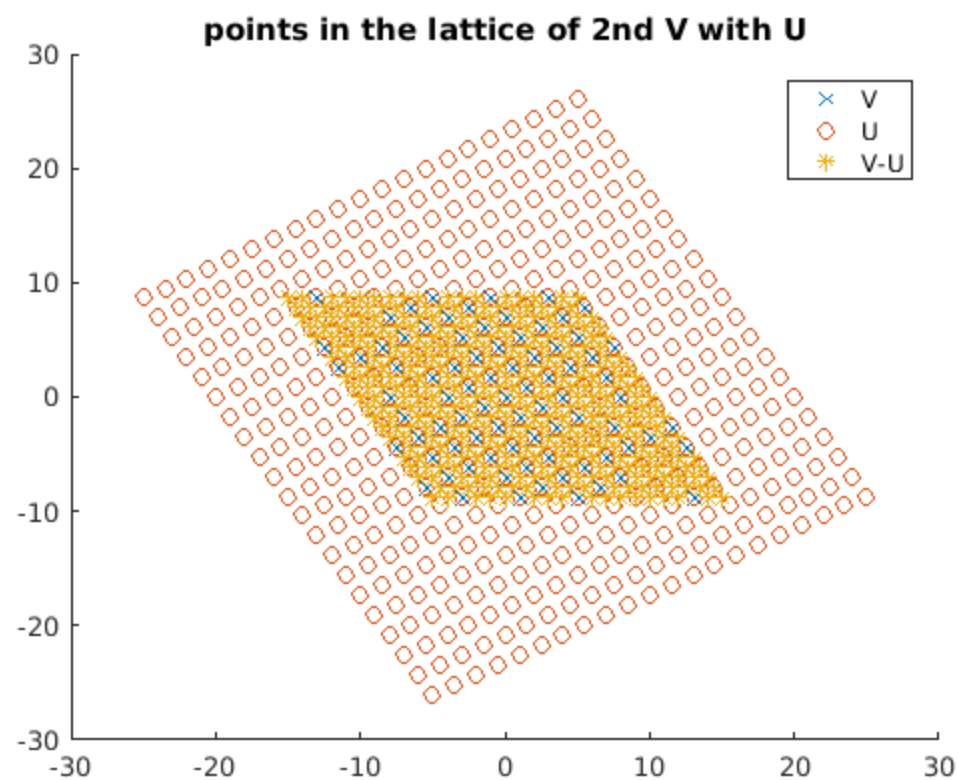
% Lu < Lv, and Lvhat < Luhut
% as for lattices V is denser than U, and for reciprocal lattices U is
% denser than V.

% In the plot you notice that some of the pixels are skipped. I think
% this
% is due to the sqrt(3) term in the lattice that triggers some
% precision
% errors.

```

---





---

## 2

on the paper

## 3

```
% I denoted Uprime and Vprime as U and V
U = [14, 2; 2, 2];
V = [3, 1; 0, 1];
M = [4, 0; 2, 2]; %overwrite M
% U == V*M % confirm that U = VM

E1 = [2,1;1,0];
mu = diag([2,4]);
E2 = [1,1;0,-1];
% M == E1*mu*E2 % confirm that the Smith form works
% det(E1);
% det(E2); % these are both -1, hence unimodular
lu = U*n_vec;
lv = V*n_vec;

figure(6);
hold on;
plot(lu(1,:),lu(2,:), 'x');
plot(lv(1,:),lv(2,:), 'o');
legend('U(prime)', 'V(prime)');
hold off;
title('Image of Lu(prime) and Lv(prime)');

% I denoted U and V to U_new and V_new
U_new = U*inv(E2);
V_new = V*E1;
% U_new == V_new*mu; %confirm that these two are equal
lu_n = U_new*n_vec;
lv_n = V_new*n_vec;

n1 = -9:2:9;
n2 = -5:4:10;
n_coset = [kron(ones(1,4),n1)',kron(n2,ones(1,10))']';
lv_n_off = V_new*n_coset;

figure(7);
hold on;
plot(lu_n(1,:),lu_n(2,:), 'o');
plot(lv_n(1,:),lv_n(2,:), 'x');
plot(lv_n_off(1,:),lv_n_off(2,:), '*');
legend('U', 'V', 'U offset [1,3]');
hold off;
title('Image of Lu and Lv (normalized)');

% The index group I(U,V) is
% {(0,1) (0,2) (0,3) (1,1) (1,2) (1,3)}
```

---

```

%Reciprocal
Vr = V_new^(-1).';
Ur = U_new^(-1).';
lur_n = Ur*n_vec;
lvr_n = Vr*n_vec;

figure(8);
hold on;
plot(lur_n(1,:),lur_n(2,:), 'x');
plot(lvr_n(1,:),lvr_n(2,:), 'o');
legend('U', 'V');
hold off;
title('Image of Lu and Lv (normalized and Reciprocal)');

%Unit Cell
% I will be doing voronoi unit cell
% By inspection the points are separated by more than 1/2. So I will
% be using a bigger window for unit cell.
%grid = -2:0.01:2-0.01;
%grid_vec = [kron(ones(1,400),grid)',kron(grid,ones(1,400))']';
adjacent_points = [0,-1,0,1;-1,0,1,0]; % group of points that are
% closest to the origin
temp_cell = V_new*adjacent_points;
vertices = temp_cell/2;
% I think it's better to just derive the unit cell from adjacent
% points
figure(9);
fill(vertices(1,:),vertices(2,:), 'b');
title('unit cell for lattice V')

temp_cell2 = Vr*adjacent_points;
vertices2 = temp_cell2/2;
figure(10);
fill(vertices2(1,:),vertices2(2,:), 'b');
title('unit cell for reciprocal lattice of V');

% Referring to previous plots, the unit cells seem to be reasonable.

% I propose
mu1 = [2,0;0,1];
mu2 = [1,0;0,4];
%mu == mu1*mu2 % they obviously match
W = V_new*mu1;
lw_n = W*n_vec;
figure(11);
hold on;
plot(lu_n(1,:),lu_n(2,:), 'x');
plot(lw_n(1,:),lw_n(2,:), 'o');
plot(lv_n(1,:),lv_n(2,:), '*');
legend('U', 'W', 'V');
hold off;
title('Image of Lu, Lv and Lw (intermediate lattice)');
% Notice that Lu < Lw < Lv

```

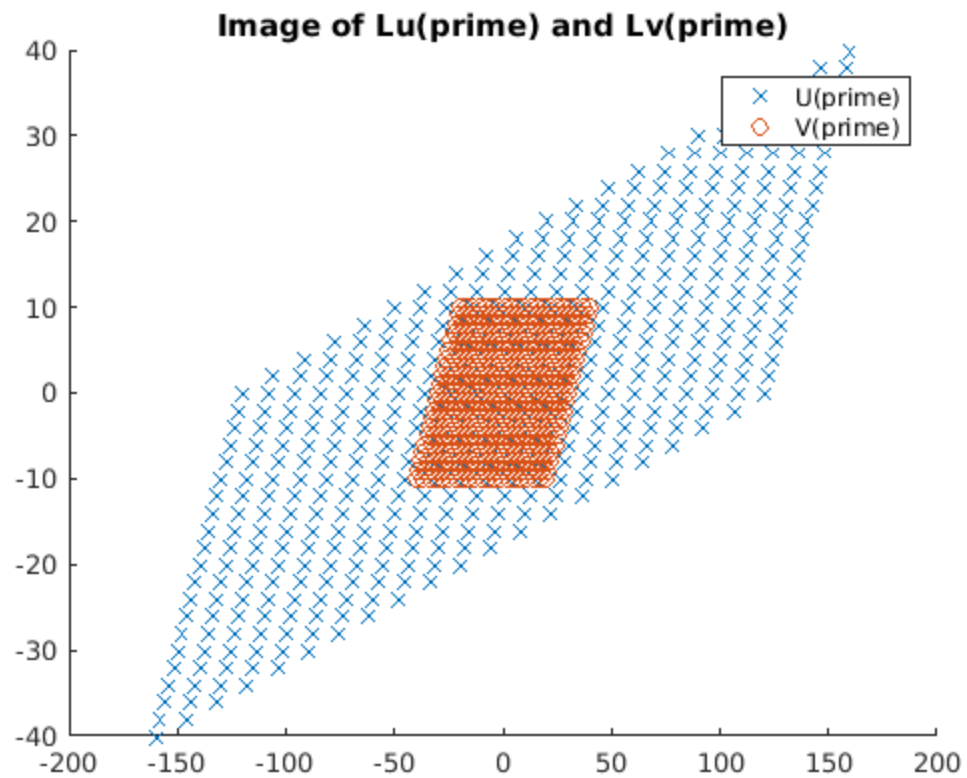
---

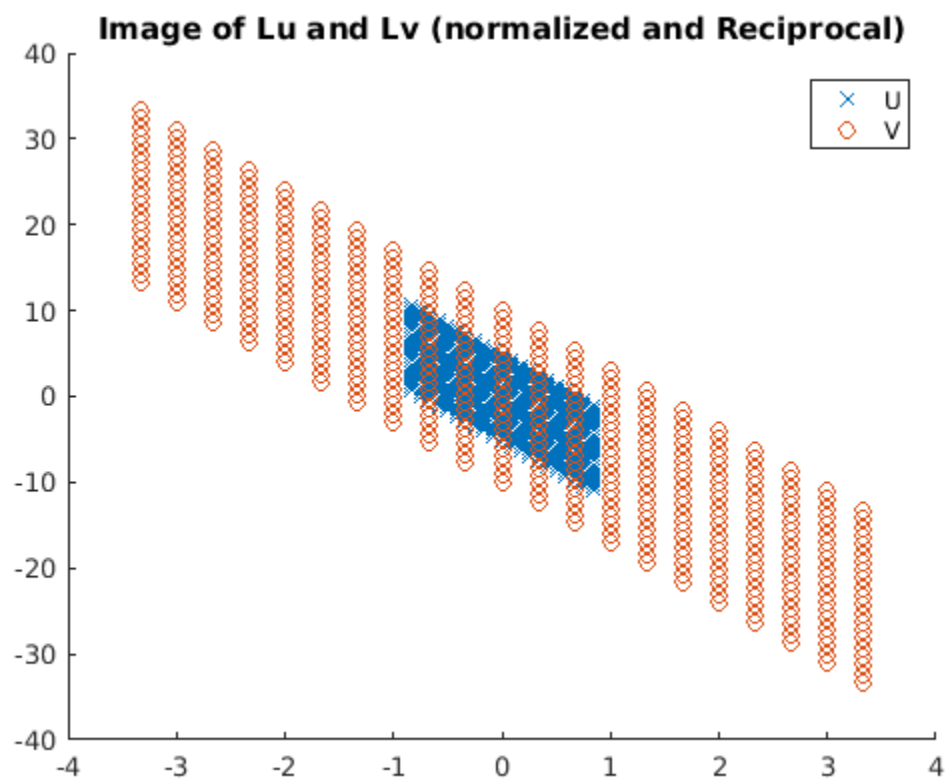
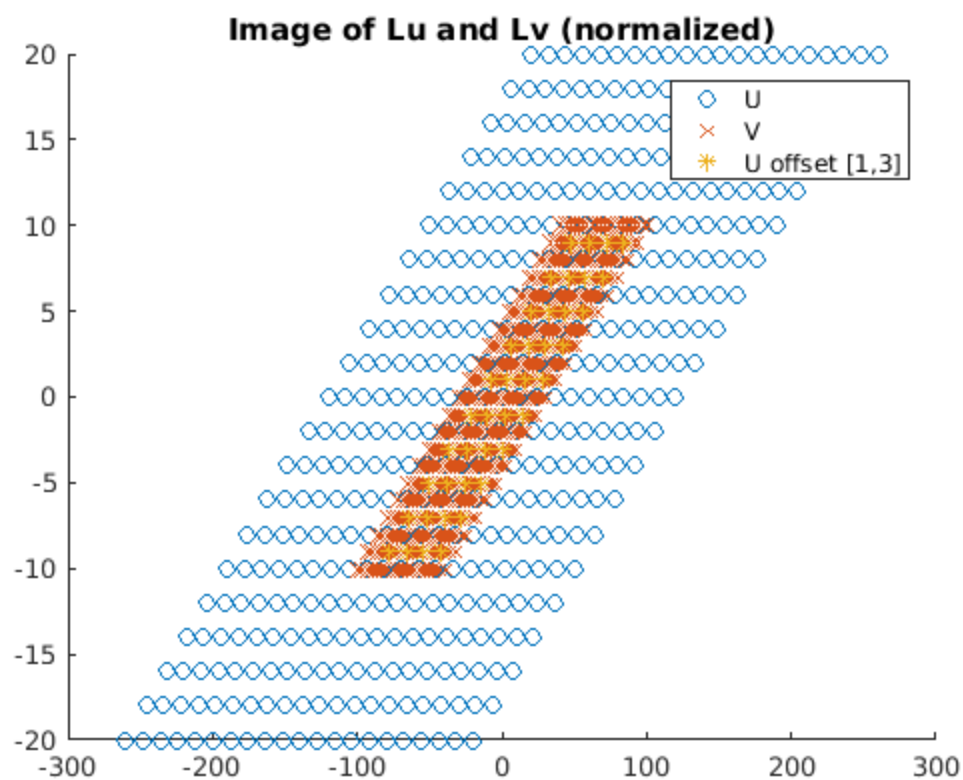
---

```

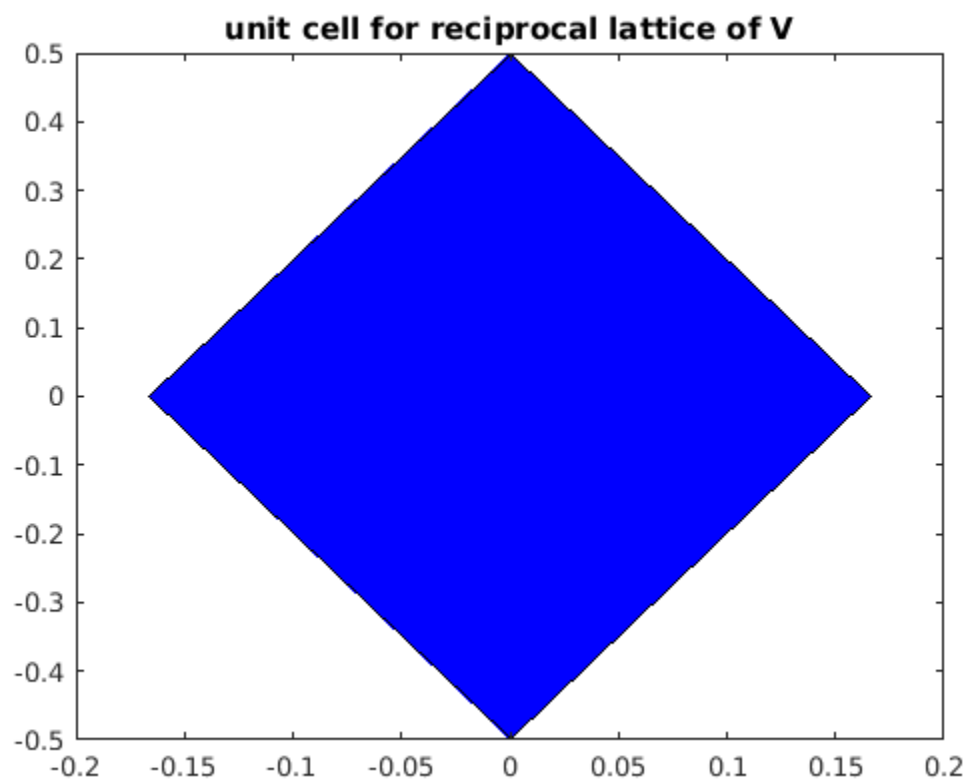
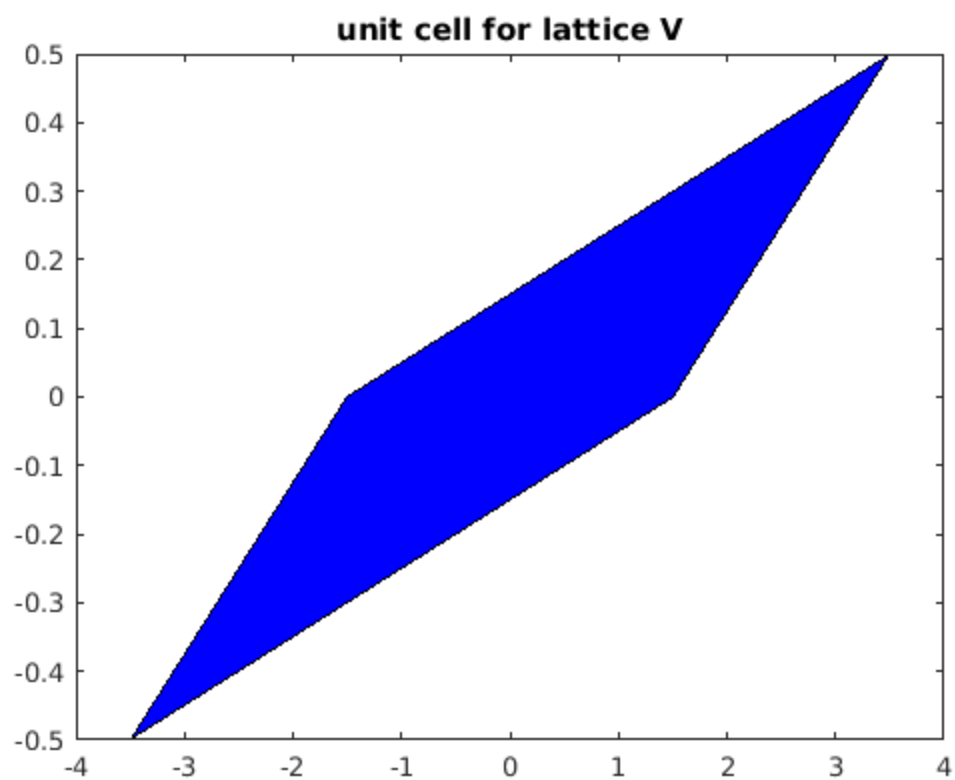
Wr = W^(-1).';
lwr_n = Wr*n_vec;
figure(12);
hold on;
plot(lur_n(1,:),lur_n(2,),'x');
plot(lwr_n(1,:),lwr_n(2,),'o');
plot(lvr_n(1,:),lvr_n(2,),'*');
legend('Ur','Wr','Vr');
hold off;
title('Image of reciprocal lattices of U, V, W(intermediate
      lattice)');
% Notiice that Lv < Lw < Lu

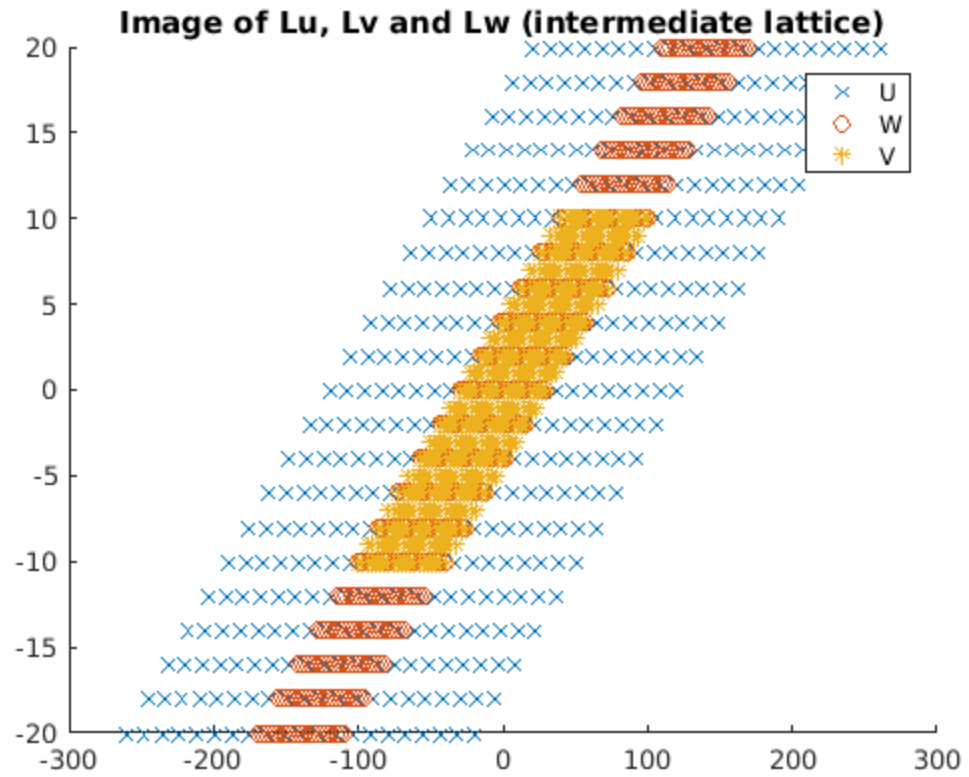
```



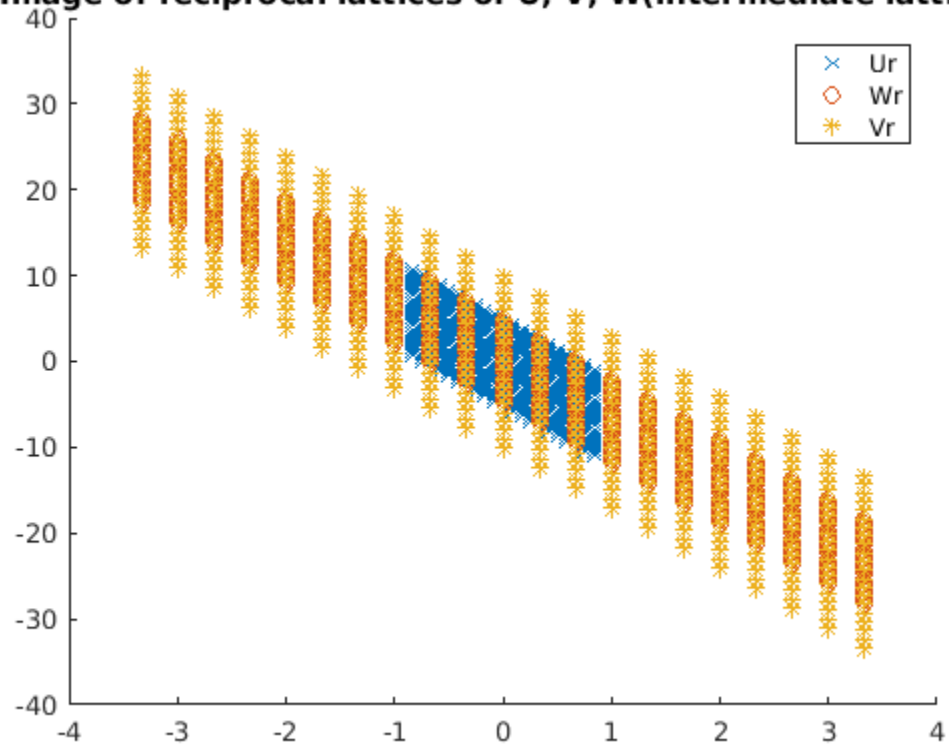








**Image of reciprocal lattices of U, V, W(intermediate lattice)**



---

## 4

functions are separate files. test functions

```
x.gen = [1,1;-1,1];
x.n = randi(10,2,20);
x.data = 1:20;

M = eye(2);
M(1) = 2; % this is like downsampling the first dimension by 2

y = decimation(x,M); % I expect data with an even number on the first
    dimension of n to survive.
disp(x)
disp(y)
y2 = interpolation(x,M); % r should be conserved
disp(x.gen*x.n)
disp(y2.gen*y2.n)
```

*y =*

*struct with fields:*

```
    n: []
  data: []
   gen: [2×2 double]

   gen: [2×2 double]
    n: [2×20 double]
  data: [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]

    n: [2×8 double]
  data: [2 5 6 7 10 14 16 20]
   gen: [2×2 double]
```

*Columns 1 through 13*

```
    19    12     8     9    20    12    15    11    15    18     8
19    15
    1     8    -6     3     0     8    -5    -7     5     2    -6
1     1
```

*Columns 14 through 20*

```
    12     9     9     4    10    11    11
   -4    -5    -7    -2     8    -3    -9
```

*Columns 1 through 13*

```
    19    12     8     9    20    12    15    11    15    18     8
19    15
```

---

	1	8	-6	3	0	8	-5	-7	5	2	-6
1	1										

Columns 14 through 20

12	9	9	4	10	11	11
-4	-5	-7	-2	8	-3	-9

## 5

```

V = V1;
V = [1 1;-1 1];
M = [2 0; 1 2];
U = V*M;

n_small = -1:1;
n_small_vec = [kron(ones(1,3),n_small)',kron(n_small,ones(1,3))'']';

x0.n = inv(V)*U*n_small_vec; % this is like my decimation function
x0.data = ones(1,9);
x0.gen = V;

%looking at M, adding 1 to all the first dimension of x0.n will move
it to a coset.
x1 = x0;
x1.n = x0.n+[1;0];

x2 = x0;
x2.n = V*n_small_vec; %from what I understand indexset I(U,V) is like
just V.

[a,b] = meshgrid(-0.5:0.01:0.49); % 100 points
X2 = zeros(100); X1 = X2; X0 = X1;

for i = 1:100
    for ii = 1:100
        f = [-0.51+i/100,-0.51+ii/100].';
        X0(i,ii)=myFT(x0,f);
        X1(i,ii)=myFT(x1,f);
        X2(i,ii)=myFT(x2,f);
    end
end

figure(13);
contour(a,b,abs(X0));
title('Magnitude Spectra of x0');

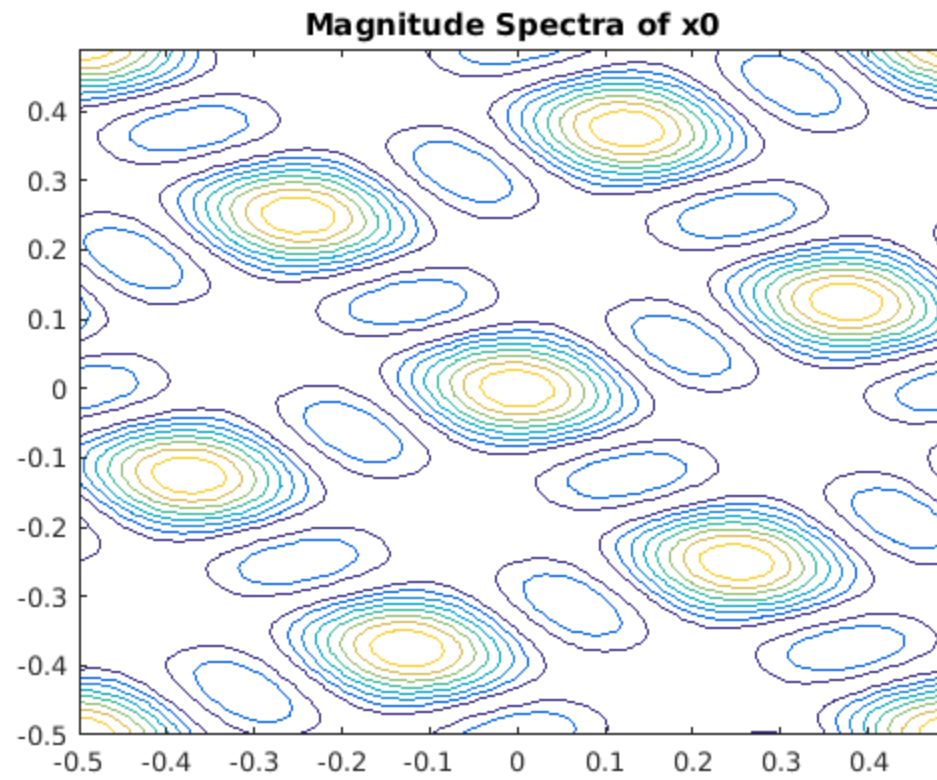
figure(14);
contour(a,b,abs(X1));
title('Magnitude Spectra of x1');

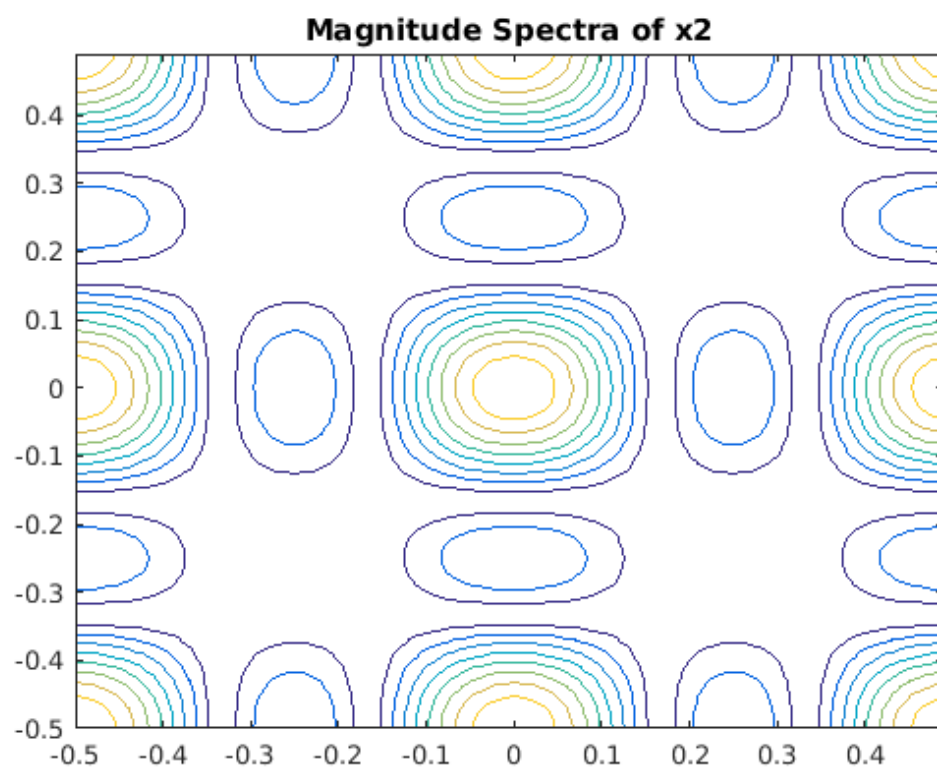
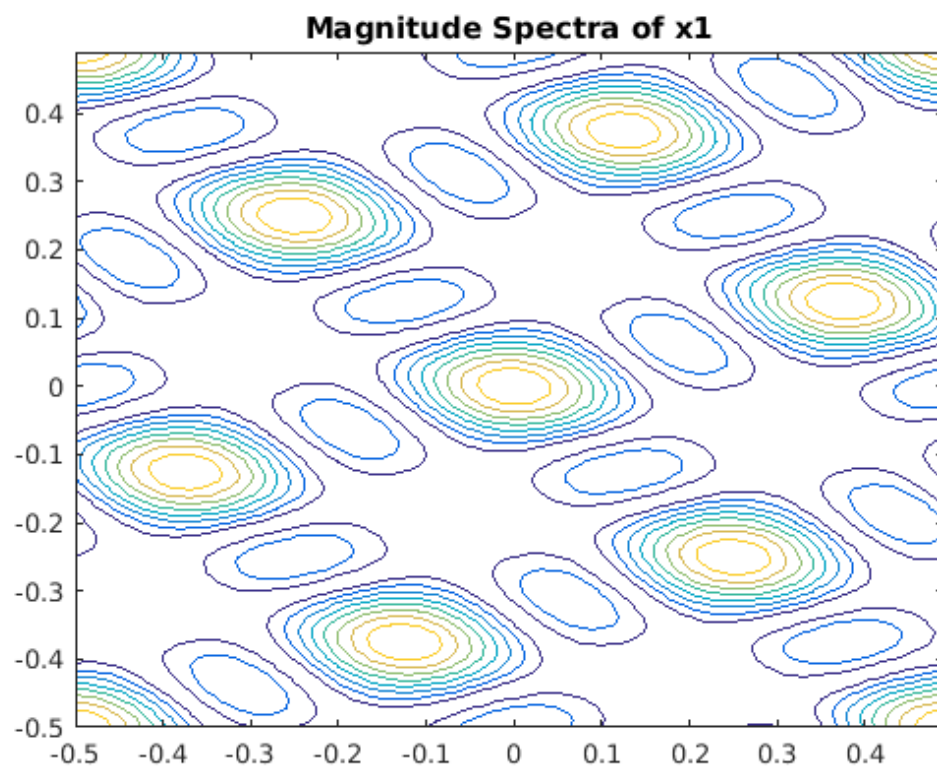
% As expected x0, x1 look the same in the magnitude spectra.

```

---

```
figure(15);  
contour(a,b,abs(X2));  
title('Magnitude Spectra of x0');
```





---

*Published with MATLAB® R2016b*