
Table of Contents

Build cov matrix, and have generate samples from it	1
Sample from correlation matrix	2
Plot all samples	3

Build cov matrix, and have generate samples from it

```
% Build the target area
boxTargetArea = boxArea( ...
    1 ,          ... wavelength
    0.5,         ... MFP
    [0,1],       ... z
    [-10,10],    ... x
    [-10,10]     ... y
);

% define HG scatter
scatter = isotropicScatter;

% make lights in far field, and views in near field
lightDirections = [0, 4];
lights = farFieldSource(deg2rad(lightDirections), 0);
views = nearFieldSource([-5;0;-10],[5;0;-10],101);
viewsPositions = views.positions(1,:);

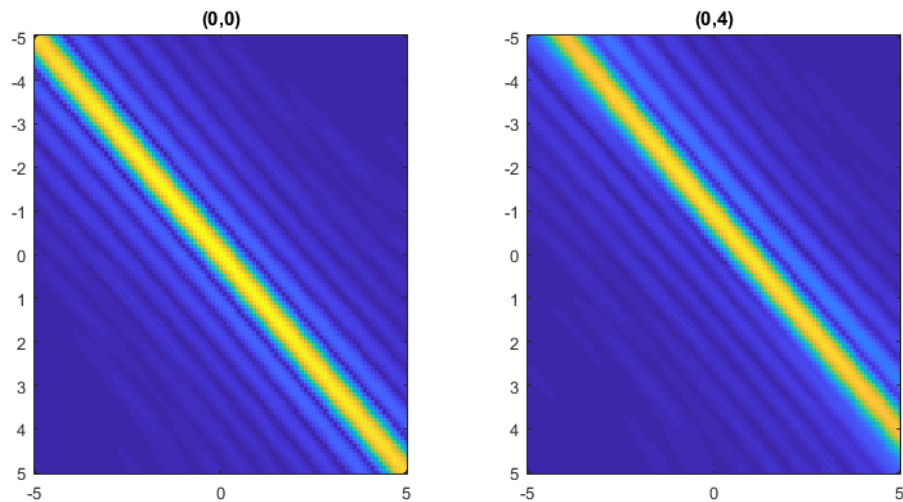
% render both the cov matrix, and one direct sample of the field
tic;
mulres = scmc(boxTargetArea, views, lights, scatter,1e3, ...
    'renderField', true, 'parforIters', 12);
toc

maxVal = max(abs(mulres.C(:)));

% show correlation matrix
f = figure;
f.Position = [0,0,870,420];
subplot(1,2,1)
imagesc(viewsPositions,viewsPositions,abs(mulres.C(:, :, 1, 1)),
[0,maxVal])
title(['(', num2str(lightDirections(1)), ', ', ...
    num2str(lightDirections(1)), ')']);

subplot(1,2,2)
imagesc(viewsPositions,viewsPositions,abs(mulres.C(:, :, 1, 2)),
[0,maxVal])
title(['(', num2str(lightDirections(1)), ', ', ...
    num2str(lightDirections(2)), ')']);
```

Elapsed time is 12.150534 seconds.



Sample from correlation matrix

Sample from complex multinormal distribution

```
% first build united C matrix
C = [mulres.C(:,:,1,1),mulres.C(:,:,1,2) ; ...
      mulres.C(:,:,2,1),mulres.C(:,:,2,2)];

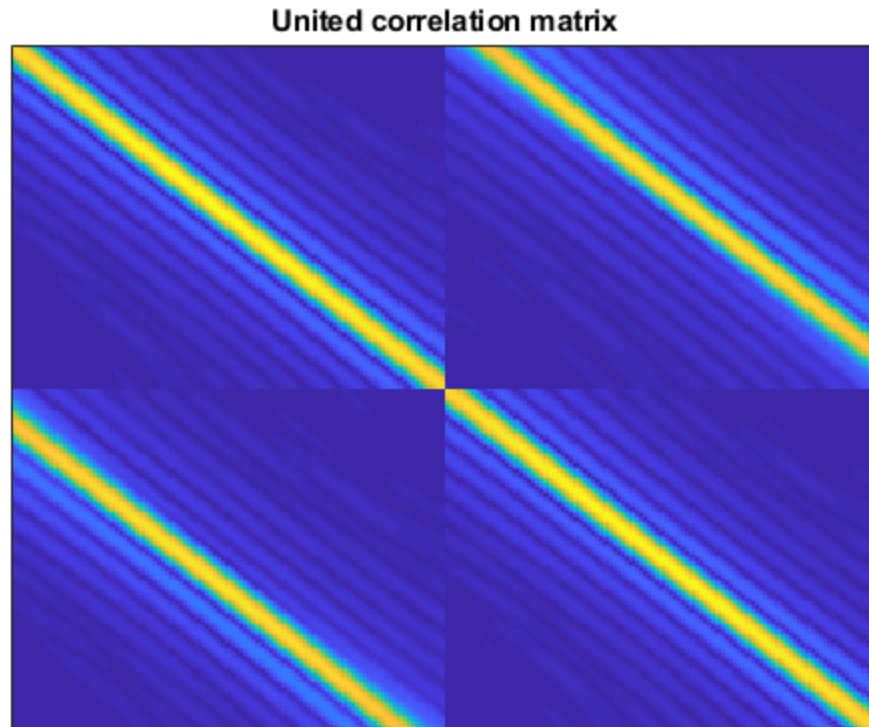
figure
imagesc(abs(C));
xticks([]);
yticks([]);
title('United correlation matrix');

% sepearate the real and complex part of the matrix
Sigma = 0.5 * [real(C), -imag(C); imag(C), real(C)];
Miu = zeros(1,size(Sigma,1));

% take two samples
sample1 = mvnrnd(Miu,Sigma);
sample2 = mvnrnd(Miu,Sigma);

% reshape to complex number
halfSample = numel(sample1)/2;
z1 = sample1(1:halfSample) + 1i * sample1(halfSample+1:end);
z2 = sample2(1:halfSample) + 1i * sample2(halfSample+1:end);

% reshape to two lighting directions
u1 = reshape(z1,[],2);
u2 = reshape(z2,[],2);
```



Plot all samples

In full lines - lighting direction of 0°

In dashed line - lighting direction of 4°

```
figure;
f = gca;
plotColors = f.ColorOrder;
hold on

l1 = plot(viewsPositions,abs(mulres.field(:,1)), ...
    'lineWidth',2,'Color',plotColors(1,:),'LineStyle','--');

plot(viewsPositions,abs(mulres.field(:,2)), ...
    'lineWidth',2,'Color',plotColors(1,:),'LineStyle','--');

l2 = plot(viewsPositions,abs(u1(:,1)), ...
    'lineWidth',2,'Color',plotColors(2,:),'LineStyle','--');

plot(viewsPositions,abs(u1(:,2)), ...
    'lineWidth',2,'Color',plotColors(2,:),'LineStyle','--');

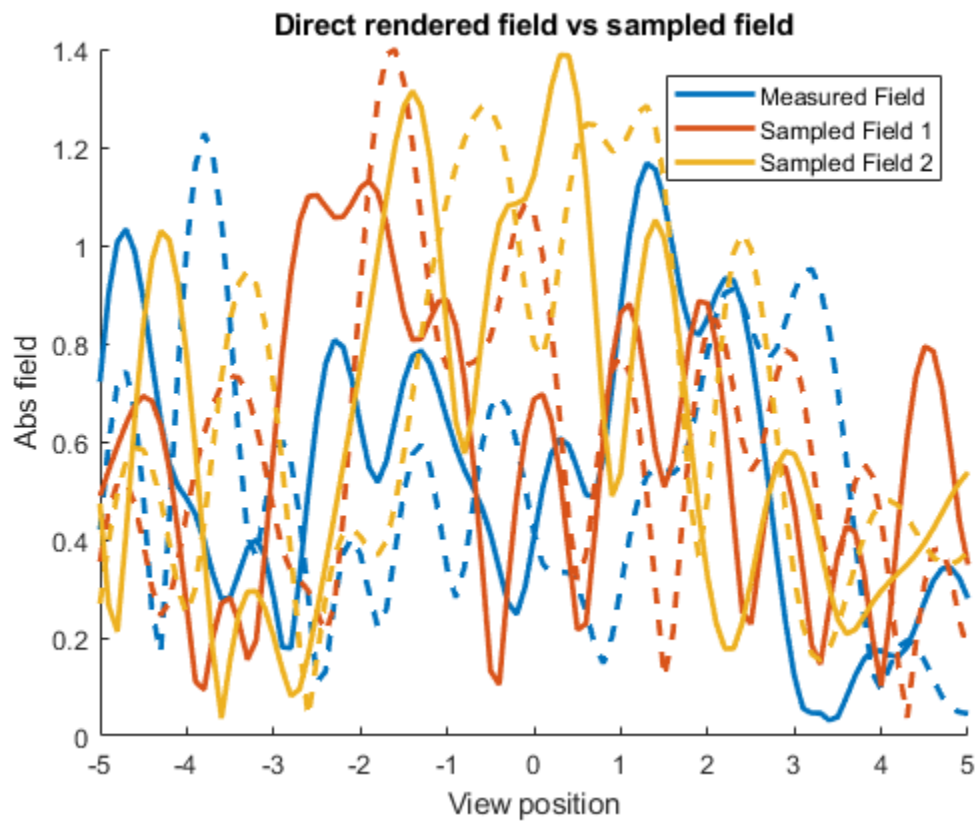
l3 = plot(viewsPositions,abs(u2(:,1)), ...
    'lineWidth',2,'Color',plotColors(3,:),'LineStyle','--');
```

```

plot(viewsPositions,abs(u2(:,2)), ...
    'linewidth',2,'Color',plotColors(3,:), 'LineStyle','--');

legend([11, 12, 13], 'Measured Field', 'Sampled Field 1', 'Sampled
    Field 2');
xlabel('View position');
ylabel('Abs field');
title('Direct rendered field vs sampled field');

```



Published with MATLAB® R2017b