```
function plot_measurements_with_uncertainty(r_meas, P_meas, lab)
%------------------------------------------------------------------
% Adds measurements and their 3D uncertainty ellipsoids to an existing figure.
%------------------------------------------------------------------
%
% Inputs:
%   r_meas : Nx3 matrix of measurement positions [x, y, z]
%   P_meas : 3x3xN covariance matrices for each measurement
%   color  : optional, color of the ellipsoids and measurement markers
%
% Example:
%   plot_measurements_with_uncertainty(r_meas, P_meas, 'r');

if nargin < 3
    lab = [];
end

color=['m'; 'y'; 'g'; 'c'];

N = size(r_meas,1);

% figure
hold on;
htemp= [];

newHandles=[];
newLabels=[];

for i = 1:N
    % Plot measurement point
    scatter3(r_meas(i,1), r_meas(i,2), r_meas(i,3), 50, color(i), 'filled');

    % Eigen decomposition to get axes of the ellipsoid
    [V, D] = eig(P_meas(:,:,i));

    % Take square root of eigenvalues
    D_sqrt = diag(sqrt(diag(D)));  % 3x3 diagonal matrix of std deviations

    % Generate a unit sphere
    [x, y, z] = ellipsoid(0, 0, 0, 1, 1, 1, 20);

    % Flatten sphere points to 3xN
    sphere_pts = [x(:)'; y(:)'; z(:)'];

    % Rotate and scale sphere to match covariance
    ellipsoid_pts = V * D_sqrt * sphere_pts;  % 3xN

    % Translate to measurement position
    x_e = reshape(ellipsoid_pts(1,:) + r_meas(i,1), size(x));
    y_e = reshape(ellipsoid_pts(2,:) + r_meas(i,2), size(y));
    z_e = reshape(ellipsoid_pts(3,:) + r_meas(i,3), size(z));

    % Plot transparent ellipsoid
    htemp(i) = surf(x_e, y_e, z_e, 'FaceColor', color(i), 'FaceAlpha', 0.2,↙
'EdgeColor', 'none');

    if nargin < 3
        newHandles = [newHandles; htemp(i)];
        newLabels = [newLabels, lab(i)];
    end
end
lgd  = legend;
existingHandles = lgd.PlotChildren;   % list of graphics handles already in legend
existingLabels  = lgd.String;
if ~(nargin <3)
    newHandles = [existingHandles(1:6); htemp(1); htemp(2); htemp(3); htemp(4)];
    newLabels  = [existingLabels(1:6), {'GS1 Uncertainty'}, {'GS2 Uncertainty'},↙
{'GPS Uncertainty'}, {'Fused Uncertainty'}];
end
legend(newHandles, newLabels);
axis equal;
xlabel('X [km]'); ylabel('Y [km]'); zlabel('Z [km]');
grid on;
hold off;

end
```