```
% basalbodynumber
function basalbody_number=basalbody_number_func(imageName,threshold)
% MATLAB version particle analysis
% input:
% image: a grey scale image: poc1B is better
% threshold: the threshold set to change the grey scale image to a
binary one
image=imread(imageName);
info=imfinfo(imageName);
if length(info)>1
    error('A grey scale image is needed')
end
if info.BitDepth~=8
    error('A 8 bit image is needed')
end
% show figure
fig1=figure(2000);clf;
subplot(2,2,1);
imshow(image);
title('red channel raw image');
axis equal;

% various ways can be used to set the range.
% method 1
% background=mean(mean(single(image)));
% noise=std(std(single(image)));
% treshold=background+4*noise;
if nargin<2
    level=graythresh(image);
else
    level=threshold/255; % for 8 bit image
end
% method 2
% method 3
% user defined level
% binary image
bw = im2bw(image,level);
figure(2000);subplot(2,2,2);
imshow(bw);
title('red channel binary image');
axis equal;
% particle properties
s=regionprops(bw, image, 'all');
% label all the clusters
figure(2000);subplot(2,2,3);
imshow(image);
title('Identified particles');
hold on
numObj = numel(s);
parameters=[];
for k = 1 : numObj
    plot(s(k).WeightedCentroid(1), s(k).WeightedCentroid(2), 'r*');
    text(s(k).WeightedCentroid(1),s(k).WeightedCentroid(2), ...
        sprintf('%4d', k), ...
        'Color','r');
    parameter=
[k,s(k).Area,s(k).Perimeter,s(k).MajorAxisLength,s(k).MinorAxisLength,
...
        s(k).Orientation];
    parameters=[parameters;parameter];
    clear parameter
end
hold off
axis equal;

% reduce the dimension by principal component analysis(PCA)
%
% set a filter
% data normalization
% paramters structure
% column 1: object number;
% column 2: object area
% column 3: object perimeter
% column 4: object major axis length
% column 5: object minor axis length
% column 6: object orientation
% column 7: object major minor axis length ratio
parameters(:,7)=parameters(:,4)./parameters(:,5);
figure(2000);subplot(2,2,4);
categories={'Area','Perimeter','Major Axis Length','Minor Axis
Length','Orientation','Major/Minor'};
boxplot(parameters(:,2:7),'orientation','horizontal','labels',categori
es);
% parameters_normalization(:,1)=parameters(:,1);
% parameters_normalization(:,2)=(parameters(:,2)-
mean(parameters(:,2)))/mean(parameters(:,2));
% parameters_normalization(:,3)=(parameters(:,3)-
mean(parameters(:,3)))/mean(parameters(:,3));
% parameters_normalization(:,4)=(parameters(:,4)-
mean(parameters(:,4)))/mean(parameters(:,4));
% parameters_normalization(:,5)=(parameters(:,5)-
mean(parameters(:,5)))/mean(parameters(:,5));
% parameters_normalization(:,6)=(parameters(:,6)-
mean(parameters(:,6)))/mean(parameters(:,6));
% parameters_normalization(:,7)=(parameters(:,7)-
mean(parameters(:,7)))/mean(parameters(:,7));
%
%
[COEFF,SCORE]=princomp(parameters_normalization(:,2:7),'VariableWeight
s','variance');
% figure()
% plot(SCORE(:,1),SCORE(:,2),'+')
% xlabel('1st Principal Component')
% ylabel('2nd Principal Component')
area_limit=quantile(parameters(:,2),0.95);

index=find(parameters(:,2)>=area_limit);
correct=round(sum(parameters(index,2)/median(parameters(:,2))))-
length(index);
particle_number_unnormalized=length(parameters);
particle_number_normalized=particle_number_unnormalized+correct;
basalbody_number=particle_number_normalized;
```

```
save([imageName(1:end-4) '_basalnumberfunction_generateddata.mat']);

% save the figure
saveas(fig1,[imageName(1:end-4) '_basalbody_number.fig'],'fig');

end
```