

Compute the depth of each point of a sample relative to its surface whatever its geometry

The following script computes the distance field between every point inside a **mask** and a **frontiere** curve*:

```
function [ebdAct] = add_dep_comp(ebsd, front_pic, mask_pic)
    %Load and adapt
    front = imread(front_pic);
    masking = imread(mask_pic);
    masking = flipdim(masking,2);
    front = flipdim(front,2);
    I_max = min([size(front,1), size(masking,1), size(ebsd,1)]);
    J_max = min([size(front,2), size(masking,2), size(ebsd,2)]);
    I_min = size(ebsd,1) - I_max+1;
    ebsd = ebsd(I_min:size(ebsd,1), 1:J_max);
    front = front(1:I_max, 1:J_max,:);
    masking = masking(1:I_max, 1:J_max,:);
    % cut frontier
    hsvValues = rgb2hsv(front);
    hueValue = hsvValues(:, :, 1);
    ebsd.prop.hue = hueValue;

    ebsdFront = ebsd(ebsd.hue>0.);

    % mask
    hsvValues2 = rgb2hsv(masking);
    hueValue2 = hsvValues2(:, :, 1);
    ebsd.prop.mask = hueValue2;

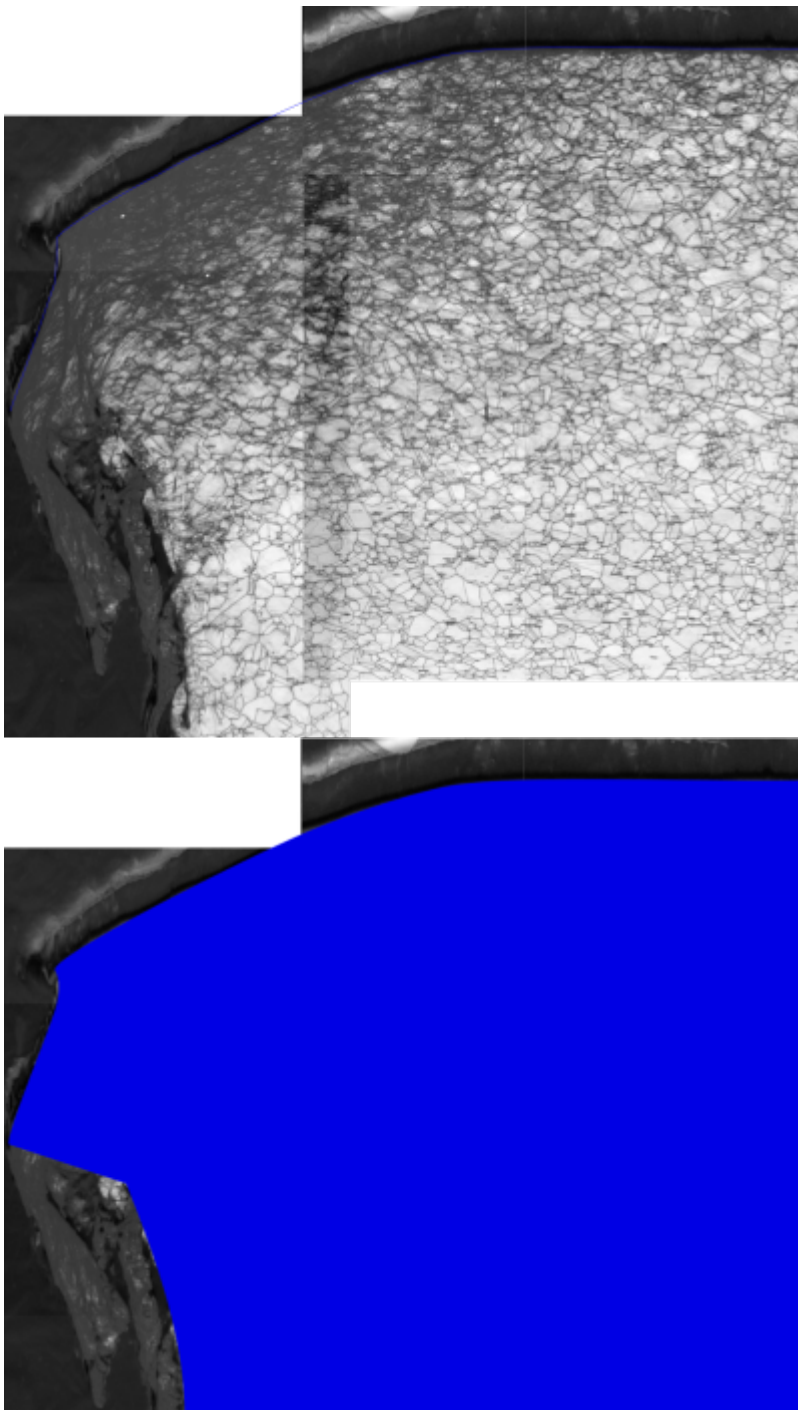
    ebsdAct = ebsd(ebsd.mask>0.);

    % Get distance with frontiere
    distTabAct = 10000*ones(ebsdAct.size);
    for i=1:ebsdFront.size(1)
        disp(strcat(string(i),"/",string(ebsdFront.size(1))))
        distTabAct = min(distTabAct,sqrt((ebsdFront.x(i)-ebsdAct.x).^2 +
(ebsdFront.y(i)-ebsdAct.y).^2));
    end
    ebsdAct.prop.dep = distTabAct;
end
```

The script takes three arguments:

- ebsd: the MTEX ebsd map object to put the distance field in. Note that the field dimensions should be the same than the picture resolution
- front_pic: A picture where everything is black and white except the frontiere line
- mask_pic: A picture where everything is black and white except the sample points used to compute the distance.

Here are two examples of picture that worked for me (left: frontiere, right: mask):



From:
<https://portail.emse.fr/dokuwiki/> - **DOC**

Permanent link:
https://portail.emse.fr/dokuwiki/doku.php?id=recherche:methodes:ebsd:ebsd_depth_map&rev=1603787243

Last update: **27/10/2020 09:27**

