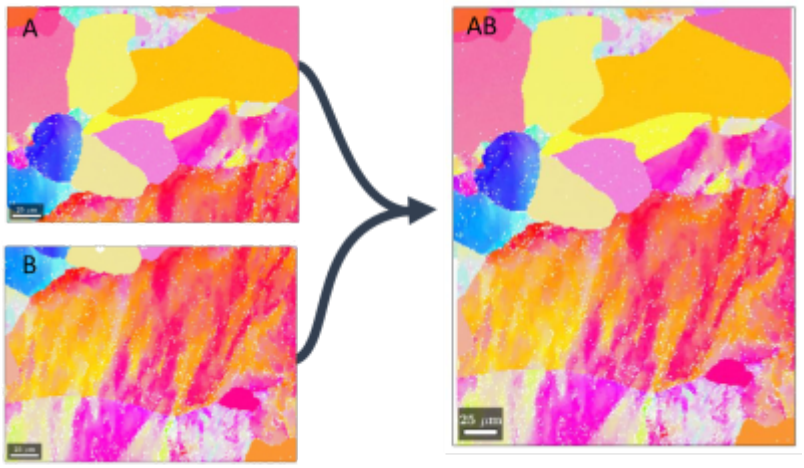
Automatic EBSD map stiching

So let's say you have two maps A and B and you want to stich them automatically to get the big map AB without any seams:



The following matlab function merges two mtex ebsd objects using the phase correlation method as explained in this presentation:

```
function [ebsd_s,t_x, t_y] = stich_map(ebsd1, ebsd2, fact, axis, phase)
    % Debruitage des carto
    ebsd1_g = fill(ebsd1(phase));
    ebsd2_g = fill(ebsd2(phase));
    % Conversion de la carto en image (nuance de gris IPFZ)
    im1 = ebsd1_g.gridify.orientations;
    im2 = ebsd2_g.gridify.orientations;
    ipfKey = ipfHSVKey(ebsd1(phase));
    [im1y, im1x] = size(im1);
    [im2y, im2x] = size(im2);
    im1 = im2uint8(rgb2gray(reshape(ipfKey.orientation2color(im1), [im1y,
im1x, 3])));
    im2 = im2uint8(rgb2gray(reshape(ipfKey.orientation2color(im2), [im2y,
im2x, 3])));
    [im1x, im1y] = invert_xy(im1x, im1y);
    [im2x, im2y] = invert_xy(im2x, im2y);
    % Calcul de la conversion pixel => coordonnees carto
    p2m_x1 = (max(ebsd1.x) - min(ebsd1.x))/im1y;
    p2m_y1 = (max(ebsd1.y) - min(ebsd1.y))/im1x;
    p2m_x2 = (max(ebsd2.x) - min(ebsd2.x))/im2y;
    p2m_y2 = (max(ebsd2.y) - min(ebsd2.y))/im2x;
    % Choix des zones de recherche
    if axis == 'y'
        taille_recouv = round(im2x/fact);
        x1d=round(im1x-2.5*taille_recouv); x1f=im1x;
        x2d=1;                    x2f=taille_recouv;
        im1p = im1(x1d:x1f,:); im2p = im2(x2d:x2f,:);
```

```
        shift11_x = 0;
        shift11_y = im1x-2.5*taille_recouv;
    elseif axis == 'x'
        taille_recouv = round(im2y/fact);
        x1d=round(im1y-2.5*taille_recouv); x1f=im1y;
        x2d=1;                     x2f=taille_recouv;
        im1p = im1(:,x1d:x1f); im2p = im2(:,x2d:x2f);
        shift11_x = im1y-2.5*taille_recouv;
        shift11_y = 0;
    else
        fprintf("Wrong axis specification ('x' or 'y' only)");
    end
    % Calcul de la transformation a appliquer
    tformEstimate21 = imregcorr(im2p,im1p,'transformtype','translation');
    t_x = (tformEstimate21.T(3,1) + shift11_x)*p2m_x1;
    t_y = (tformEstimate21.T(3,2)+ shift11_y)*p2m_y1;
    %Fusion des objets EBSD
    ebsd_t = ebsd2;
    ebsd_t.x = ebsd_t.x + t_x;
    ebsd_t.y = ebsd_t.y + t_y;
    ebsd_t = ebsd_t(ebsd_t.y > max(ebsd1.y) | ebsd_t.x > max(ebsd1.x));
    ebsd_s = [ebsd1 ebsd_t];

end
```

The function takes the following entry parameters:

- **ebsd1, ebsd2**: the two ebsd object to stich (2 must be at the right of 1 for the x axis, or at the bottom of 1 for the y axis)
- **fact**: the length ratio of the zone to use in image 2 for the phase correlation method
- **axis**: chose either to stich map 1 on top of map 2 or map1 at the left of map 2 (could be generalized I guess..)
- **phase**: the phase to use to generate the orientation field used for stiching

Here's an example using the function to automatically stich several maps along the y and x axis successively. The scripts pauses after each column and after stiching two columns to ask if the stiching is correct. Is not, the **fact** parameter can be changed until it works.

```
clear
close all
clc


%% Choix des paramètres

% Nom des images
carto1 = 'carto/1/Echantillon L14 1800C 100s grande zone_0_0.crc';
preset = 'carto/1/Echantillon L14 1800C 100s grande zone_';
```

```
ix_max = 7;
iy_max = 7;
verif=0;
phase = 'Tungstène';

%%

CS = {...
  'notIndexed',...
  crystalSymmetry('m-3m', [3.2 3.2 3.2], 'mineral', phase, 'color', [0.53
0.81 0.98])};

% plotting convention
setMTEXpref('xAxisDirection','east');
setMTEXpref('zAxisDirection','intoPlane');



f = waitbar(0,'1','Name','Stiching maps...',...
    'CreateCancelBtn','setappdata(gcbf,''canceling'',1)');
setappdata(f,'canceling',0);

factx = 7.;
facty = 7.;
diffx = 0;
diffy = 0;
for j=0:iy_max
    done = 0;
    while ~done
        ebsd1 =
EBSD.load(carto1,CS,'interface','crc','convertEuler2SpatialReferenceFrame');
        ebsd_s = ebsd1;
        for i=0:ix_max
            waitbar((j*ix_max+i)/((ix_max+1)*(iy_max+1)),f,"Stitching
maps...");
            if getappdata(f,'canceling')
                break
            end

            if ~(i==0)
                carto_actu = [preset num2str(j) '_' num2str(i) '.crc'];
                ebsd2 =
EBSD.load(carto_actu,CS,'interface','crc','convertEuler2SpatialReferenceFram
e');
                [ebsd_s, tx, ty] = stich_map(ebsd_s, ebsd2, factx, 'y',
phase);
            end
        end
        plot(ebsd_s(phase), ebsd_s(phase).orientations)
        done = input("Is the column correct ? (0=no - 1=yes)");
        if ~done
```

```
            factx = input("Specify a new covering zone parameter: ");
        end
    end
    if j==0
        ebsd_final = ebsd_s;
    else
        done = 0;
        while ~done
            carto_j_0 = [preset num2str(j-1) '_0.crc'];
            carto_jp1_0 = [preset num2str(j) '_0.crc'];
            ebsd_j_0 =
EBSD.load(carto_j_0,CS,'interface','crc','convertEuler2SpatialReferenceFrame
');
            ebsd_jp1_0 =
EBSD.load(carto_jp1_0,CS,'interface','crc','convertEuler2SpatialReferenceFra
me');
            [tmp, tx, ty] = stich_map(ebsd_j_0, ebsd_jp1_0, facty, 'x',
phase);
            ebsd_s_tmp = ebsd_s;
            ebsd_s_tmp.x = ebsd_s_tmp.x + tx + diffx;
            ebsd_s_tmp.y = ebsd_s_tmp.y + ty + diffy;
            ebsd_f_t = [ebsd_final ebsd_s_tmp];
            plot(ebsd_f_t(phase), ebsd_f_t(phase).orientations)
            done = input("Is the new global map correct ? (0=no - 1=yes)");
            if ~done
               facty = input("Specify a new covering zone parameter: ");
            end
        end
        diffx = diffx + tx;
        diffy = diffy + ty;
        ebsd_final = ebsd_f_t;
%         ebsd_final.x = ebsd_final.x - min(ebsd_final.x);
%         ebsd_final.y = ebsd_final.y - min(ebsd_final.y);
    end
    carto1 = [preset num2str(j+1) '_0.crc'];
end

delete(f)

ebsd_final.export("Carto_global.ctf")
```