1 Matlab correction

1.1 Main function

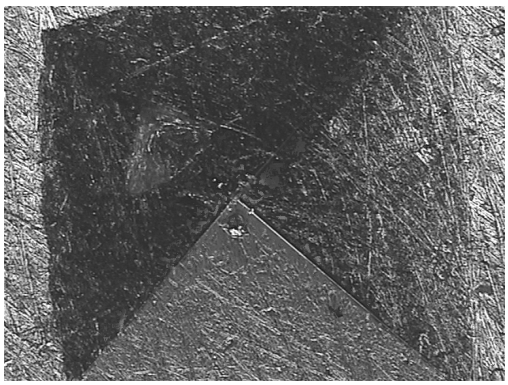
This function is used as the main function to perform shape-from-focus reconstruction. The parameter method is the name of the function. Notice that tha variable stack is of type double to handle the cornea stack (which is a 16 bits image).

```
function [Z, T]=SFF(file, N, method)
% Shape From Focus by SML method
% Z: heights/altitudes (indices)
% T: texture
%
% file: filename (tif stack-file)
% N: parameter of 'method' function
% method: method to use (SML, variance, etc.)
info = imfinfo(file);
num_images = numel(info);

% create stack into memory
stackF=zeros(info(1).Height, info(1).Width, num_images);

% load stack
stack =double(stackF);
for k = 1:num_images
    stack(:,:,k)=imread(file, k);
    % ... compute SML
    stackF(:,:,k) = feval(method, double(stack(:,:,k)), N);
end

% search for maximum of function
[~, Z] = max(stackF, [], 3);
Z = uint8(Z);
T = double(zeros(size(stack,1), size(stack,2)));
for i=1:size(T, 1)
    for j=1:size(T,2)
        T(i,j) = stack(i,j,Z(i,j));
    end
end

figure();
subplot(1,2,1); imshow(Z,[]); title('altitudes');
subplot(1,2,2); imshow(T,[]); title('textures');
end
```

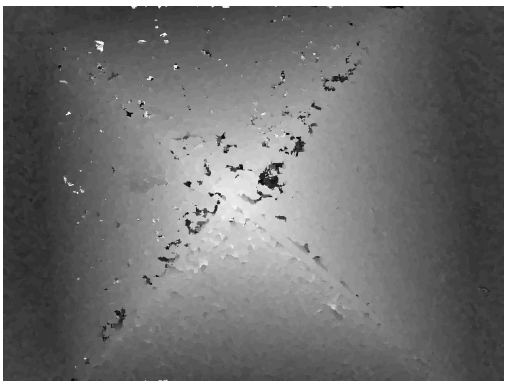
1.2 Sum of Modified Laplacian

Results are illustrated in Fig.1.

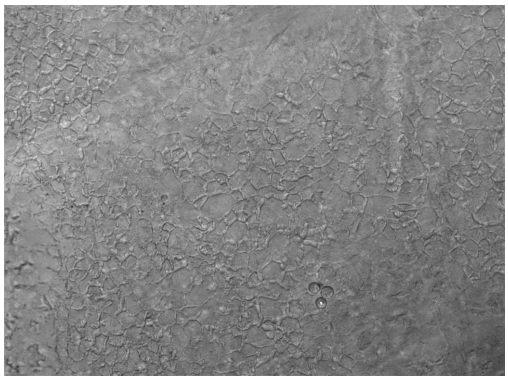
```
% main function
function SML=modifiedLaplacian(A, N)
    h1 = [0 0 0; -1 2 -1; 0 0 0];
    h2 = h1';
    ML = abs(conv2(A, h1, 'same')) + abs(conv2(A, h2, 'same'));

    h = ones(N);
    SML = conv2(ML, h, 'same');
end
```



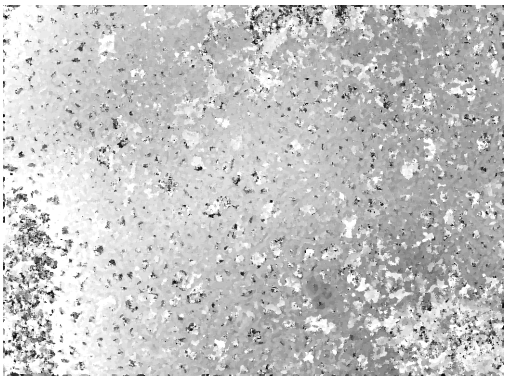
(a) Texture.



(b) Altitudes.



(c) Texture.

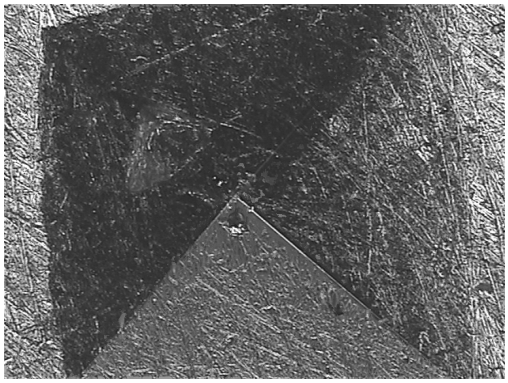


(d) Altitudes.

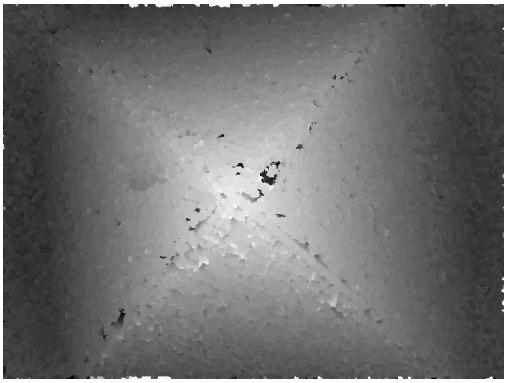
Figure 1: Texture and altitude reconstruction with the SML method.

1.3 Variance

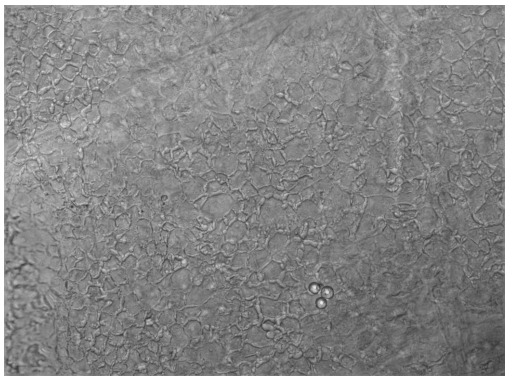
The focus measure based on the variance i a really simple method that works in most cases, see Fig.2.



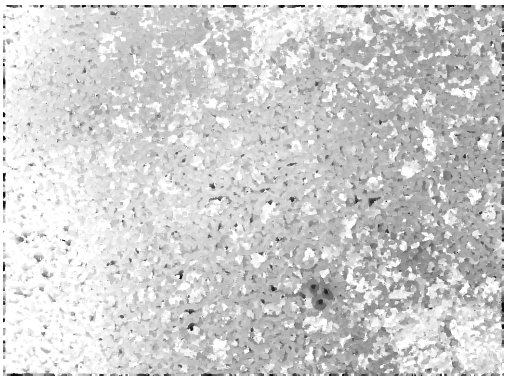
(a) Texture.



(b) Altitudes.



(c) Texture.



(d) Altitudes.

Figure 2: Texture and altitude reconstruction with the variance method.

```
function V=variance(A, N)
% A: single image
% N: size of the window
h = ones(N);

moyenne = (1/N^2) * conv2(A, h, 'same');
D2=(A-moyenne).^2;
% variance
V = conv2(D2, h, 'same');
end
```

1.4 Tenengrad

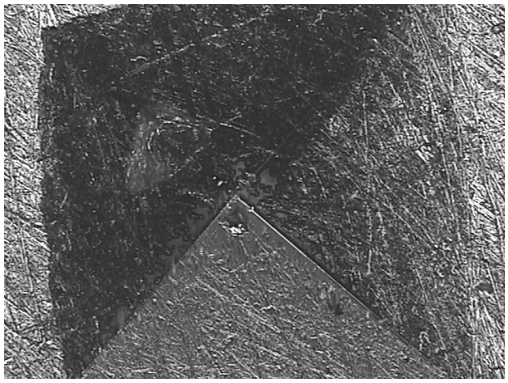
The tenengrad method is base on a Sobel filter, see Fig.3.

```
function T=tenengrad(A, ~)
% A: single image
    Sx = fspecial('sobel');
    Gx = imfilter(double(A), Sx, 'replicate', 'conv');
    Gy = imfilter(double(A), Sx', 'replicate', 'conv');
    T = Gx.^2 + Gy.^2;
end
```

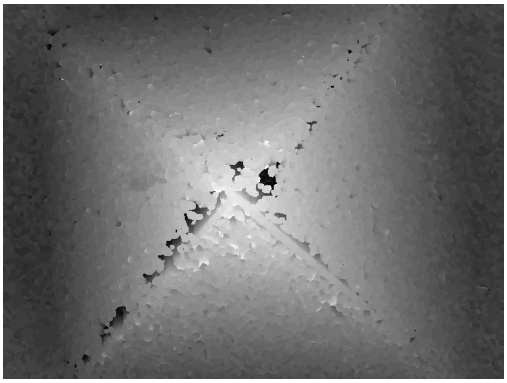
1.5 Variance of Tenengrad

The variance of Tenengrad is an improvement of the Tenengrad method, see Fig.4.

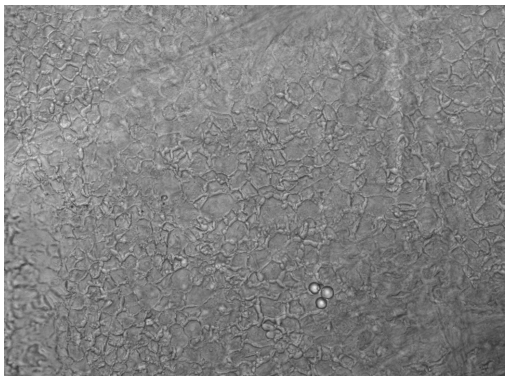
```
function vt=varianceTenengrad(A, N)
    h1 = [1 2 1; 0 0 0; -1 -2 -1];
    h2 = h1';
    ML = sqrt(conv2(A, h1, 'same').^2 + conv2(A, h2, 'same').^2);

    vt=variance(ML, N); % variance function previously defined
end
```



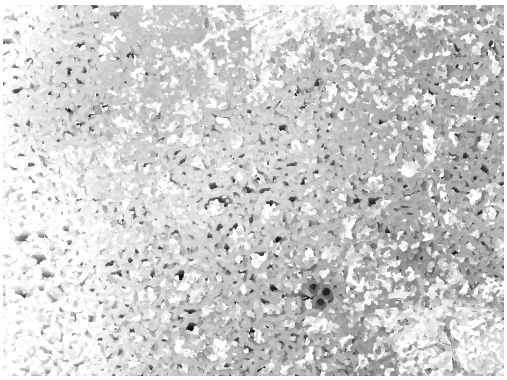
(a) Texture.



(b) Altitudes.

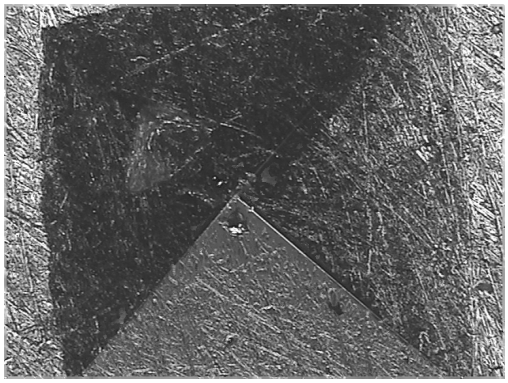


(c) Texture.

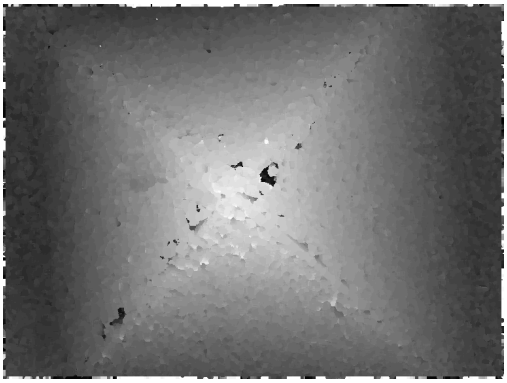


(d) Altitudes.

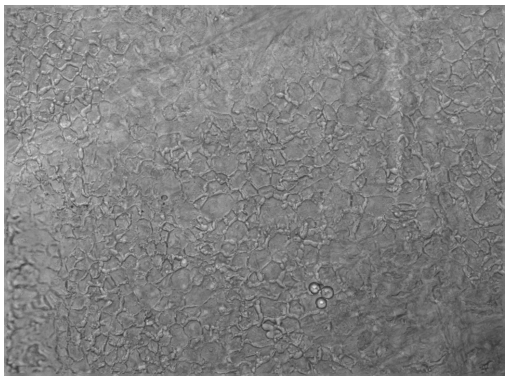
Figure 3: Texture and altitude reconstruction with the SML method.



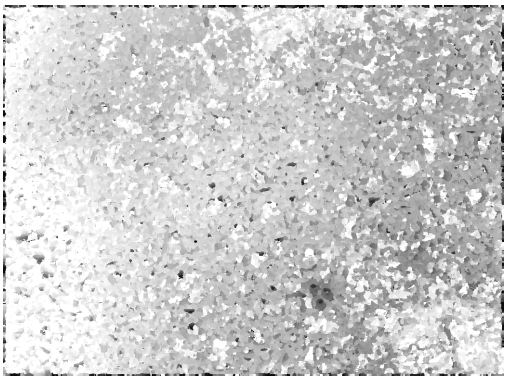
(a) Texture.



(b) Altitudes.



(c) Texture.



(d) Altitudes.

Figure 4: Texture and altitude reconstruction with the variance of Tenengrad method.