Correction: Tutorial Logarithmic Image Processing (LIP)

1 Matlab correction

1.1 Elementary operations

The most important function to code is the graytone transformation function. It considers M as the maximum value (the absolute white). This code means that the absolute white cannot be reached, and thus $f \in [0; M[$. After discretizing the gray values, $F \in [0; M-1]$.

```
function f=graytone(F, M)
% graytone transform
% this is the most important
f = M-eps(M)-F;
```

```
function l=phi(f, M)
% isomorphism
l = -M*log(1-f/M);
end

function f=invphi(l)
f = M*(1-exp(-l/M));
end
```

```
function z=plusLIP(x,y,M)
a=double(x);
b=double(y);
z=a+b-a.*b/M;
```

```
function z=timesLIP(alpha,x,M)
a=double(x);
z=M-M*(1-a/M).^alpha;
```

1.2 LIP dynamic expansion

The optimal value for dynamic expansion is given by λ_0:

$$\lambda_0(f) = \arg\max_{\lambda} \{\max(\lambda \mathbin{\triangle\!\!\!\!\times} f) - \min(\lambda \mathbin{\triangle\!\!\!\!\times} f)\}$$

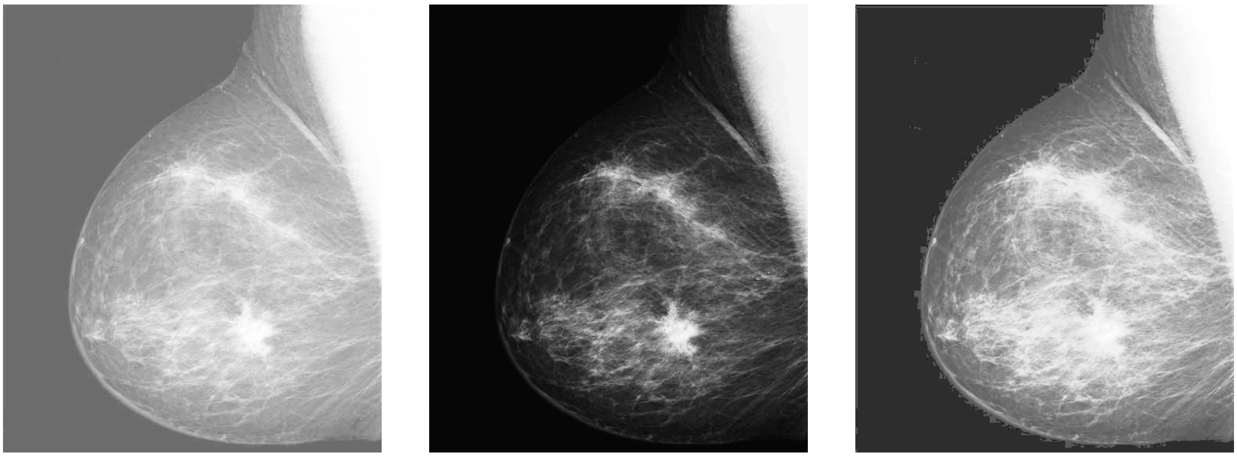
Let $A(\lambda) = \max(\lambda \mathbin{\triangle\!\!\!\!\times} f) - \min(\lambda \mathbin{\triangle\!\!\!\!\times} f) = \lambda \mathbin{\triangle\!\!\!\!\times} \max(f) - \lambda \mathbin{\triangle\!\!\!\!\times} \min(f)$ and $B = \ln(1 - \min(f)/M)$ et $C = \ln(1 - \max(f)/M)$.

$$\begin{aligned}
A'(\lambda) = 0 \quad &\Leftarrow \quad [(M - M\exp(\lambda C)) - (M - M\exp(\lambda B))]' = 0 \\
&\Leftarrow \quad [\exp(\lambda B) - \exp(\lambda C)]' = 0 \\
&\Leftarrow \quad B\exp(\lambda B) - C\exp(\lambda C) = 0 \\
&\Leftarrow \quad \ln(B) + \lambda B = \ln(C) - \lambda C \\
&\Leftarrow \quad \lambda = \frac{\ln(C) - \ln(B)}{B - C} \\
&\Leftarrow \quad \lambda = \frac{\ln(C/B)}{B - C}
\end{aligned}$$

Thus, yielding to:

$$\lambda_0(f) = \frac{\ln\left(\dfrac{\ln(1 - \max(f)/M)}{\ln(1 - \min(f)/M)}\right)}{\ln\left(\dfrac{M - \min(f)}{M - \max(f)}\right)}$$

The results are shown in Fig. 1.



(a) Original image.

(b) LIP dynamic expansion.

(c) Histogram equalization.

Figure 1: Results of dynamic expansion. The dark part of the image appears darker.

1.3 Edge detection

When applying an operator in the LIP framework, it is better to apply first the isomorphisme, then the operator, and then get back into the classical space. This is applied for example for an edge detection operator.

1.4 Complete matlab code

```matlab
%% 1 - Elementary LIP operations
M=256;

% read image
B=imread('breast.tif');
B=double(B);
% show image
figure;viewImage(B);title('Image originale');
% gray-tone function
tone = graytone(B, M);

D=graytone(timesLIP(2, tone), M);
figure;imshow(D,[0 256]);title('LIP scalar multiplication by 2');

%% 2 - Dynamic expansion

l=computeLambda(tone)
E=graytone(timesLIP(l,tone), M);
figure
subplot(1,3,1);imshow(B, [0 256]);title('Original image');
subplot(1,3,2);imshow(E,[0 256]);title('Dynamic expansion');
imwrite(uint8(E), 'expanded.png');
subplot(1,3,3);imshow(255*histeq(B/255),[0 256]);
title('Histogram equalization');
imwrite(uint8(255*histeq(B/255)), 'histeq.png');

%% 3 - Contour detection
% The results are not really convincing in this case.
% The important thing is the use of the isomorphism to simplify the
% computations.
BW = edge(phi(tone, M));
figure(); imshow(BW); title('LIP edge detection');

BW = edge(B);
figure(); imshow(BW); title('Classic edge detection');
```