

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Information Entropy

Jan Urban

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63401>

Abstract

“The chapter begins with the short description about the concept of entropy, formula, and matlab code. Within the main chapter body, three different approaches how to use the information entropy in dataset analysis: (i) for data segmentation into two groups; (ii) for filtration of the noise in the dataset; (iii) for enhancement of the entropy contribution via point information gain. Finally, the conclusion is briefly about extended analysis using more generalized entropy, and the usability of described algorithms: advantages and disadvantages.”

Keywords: information, entropy, Shannon, segmentation, thresholding, filtration, point information gain

1. Introduction

MATLAB environment enables advanced data processing and analysis, especially using its toolboxes like signal processing, image processing, and statistics.¹ The real signals have to be evaluated with numerous methods for filtration, transformation, alignment, comparison, and so on to extract the hidden knowledge. These methods are belonging to the large group of data processing and analysis. Their origin is different from statistics, physics, artificial intelligence, or systems theory. Recently, Katajama [1] pronounced a clear distinction between the processing and analysis (**Figures 1 and 2**).

Processing

¹ MATLAB and its toolboxes are trademarks or registered trademarks of The MathWorks, Inc.

- is the necessary step before the analysis.
- transforms the raw data into more transparent format for the analysis.
- includes tasks as calibration, filtering, feature detection, alignment, normalization, modeling, and so on.

Analysis

- is the interpretation of the processed data.
- consists of comparison, classification, clustering, decomposition, pattern recognition, identification, and so on.

One of the most useful plots in the signal or image analysis is the signal histogram, an expression of signal abundance, first introduced by Pearson [2]. The estimation of proper histogram, as a representation of the probability distribution function, suffers with the question of the proper binning. However, in the digital era, we are live with the datasets, which are discrete representation of discrete events of the real signal. Thus, the amount of bins is usually given by the amount of quantization levels during the sampling process (**Figures 3 and 4**).

In this chapter, the question of image processing is discussed. The lecture opens the intensity histogram function, and the induction continues through the statistical parameters, like central moments, to the information entropy. Three different methods for using the entropy in image processing are introduced, entropy filtration, entropy segmentation, and point information gain. The description is completed by mathematical equations as well as by commented MATLAB commands. The results of the commands are the plots and figures presented within the text. This chapter aims to serve as guiding overview for the entropy consideration as a processing method. The simple examples show the methods steps and additional features (**Figure 5**).

2. Histogram function

In digital image representation, intensity histogram $H(p)$ of a grayscale image is an intensity function shows count of pixel $\Phi(i, j)$ with the intensity equals d independently on the position (i, j) :

$$\begin{aligned}
 H(d) &= \sum_{i,j} h(i, j, d); \\
 h(i, j, d) &= 1, \text{ if } \Phi(i, j) = d; \\
 &= 0, \text{ if } \Phi(i, j) \neq d;
 \end{aligned}
 \tag{1}$$

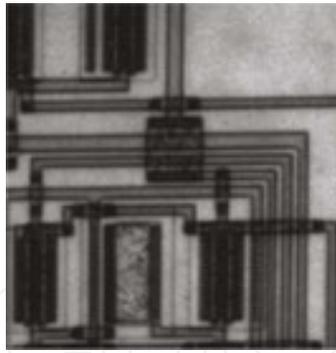


Figure 1. Image of circuit.

In MATLAB, the grayscale image *circuit*² could be loaded by the following commands:

```
Im = imread('circuit.tif'); %load the image;  
Im = double (Im) / 255; %convert to double, range 0-1;  
figure, imshow (Im); %show the image;  
title ('Image of circuit'); %add caption to the image;
```

Do not forget the ';' symbol at the end of each command, otherwise MATLAB will write all the pixels values!

The image is loaded into the variable *Im*. Its intensity levels are between zero and 255 (8-bit coding) with single precision. The second command covers the image *Im* into double precision and rescale the intensity values for the range from zero to one. The size of the image is $M \times N$, which is also the amount of the pixels in the image.

$$\begin{aligned} i &\in \langle 0; M \rangle; \\ j &\in \langle 0; N \rangle; \\ \sum_{i,j} &= \sum_{0,0}^{M,N}; \\ d &\in \langle 0,1 \rangle; \end{aligned} \quad (2)$$

To compute the histogram:

```
[H,d] = imhist (Im); %compute histogram of the image;  
figure, plot (d, H); %show the histogram function;  
xlabel ('d'); %add caption to the x-axis;  
ylabel ('H'); %add caption to the y-axis;
```

² MathWorks builtin demo image pre-packaged into MATLAB.

```
title ('Histogram of circuit'); %add caption to the plot;
```

where the variable H is the histogram function, and variable d are the d values (intensity levels). The H is a function of d .

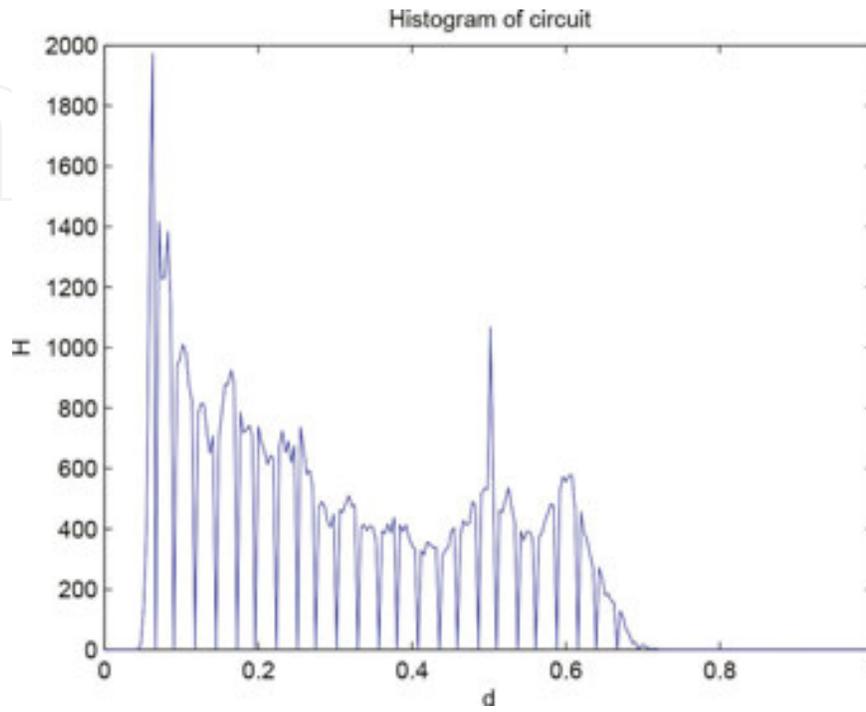


Figure 2. Histogram of circuit.

The histogram represents almost the distribution of the values in the image. To obtain the estimation of the distribution function, it is necessary to normalize the histogram function H to sum equals one³:

$$\begin{aligned} MN &= \sum_d H; \\ H &= \frac{H}{MN}; \end{aligned} \quad (3)$$

```
MN = sum (H); %count histogram area;
```

```
H = H./MN; %normalize histogram;
```

```
figure, plot (d,H); %show normalized histogram;
```

```
xlabel ('d');
```

³ The 0th moment: $\mu_0 = E[(D - E[D])^0] = \sum_i d_k^0 p(d_k) = \sum_k p(d_k)$. The fact that the probability distribution of D is normalized means that the 0th moment is always 1.

```
ylabel ('probability');
title ('Probability of circuit');
```

This could be used for many modification (contrast enhancement, frequency evaluation, segmentation, ..., etc.).

2.1. Statistical parameters

The distribution function allows us to compute some statistical parameters relevant for the further processing. The distribution is well characterized by two parameters, the location parameter and scaling parameter. The location parameter describes the value around which are all other values.

$$\mu = \frac{\sum_k H_k * d_k}{\sum H_k} \quad (4)$$

```
%compute mean value from the histogram;
mu = sum (H.*d) / sum(H)
```

This time, do not write the * ; ' symbol at the end of the sentence, to see the result of calculation. This value express the position on the x -axis (d-value), around which is the distribution centered. The value means weighted arithmetic average of all intensity levels. The levels are weighted according to the probability estimated from the histogram function H ; The value of the weighted arithmetic average is called mean value⁴ μ . Mean is also the first central moment.

```
figure, plot (d,H);
xlabel ('d');
ylabel ('probability');
title ('Probability of circuit and mean value');
```

```
%add mean value to the probability plot;
hold on, plot ([mu, mu], [0, max(H)], 'r');
```

There is another way how to obtain the mean value directly from the intensity levels of all pixels in the image Im .

```
%compute mean value from the image;
mu = mean ( reshape ( Im,size (Im,1)*size (Im,2),1 ) )
```

The image is reshaped into vector of the size $[M \times N, 1]$. No information is lost, only some computations will be simpler to proceed (**Figures 6 and 7**).

⁴ There are three Pythagorean means: Arithmetic, Geometric, and Harmonic

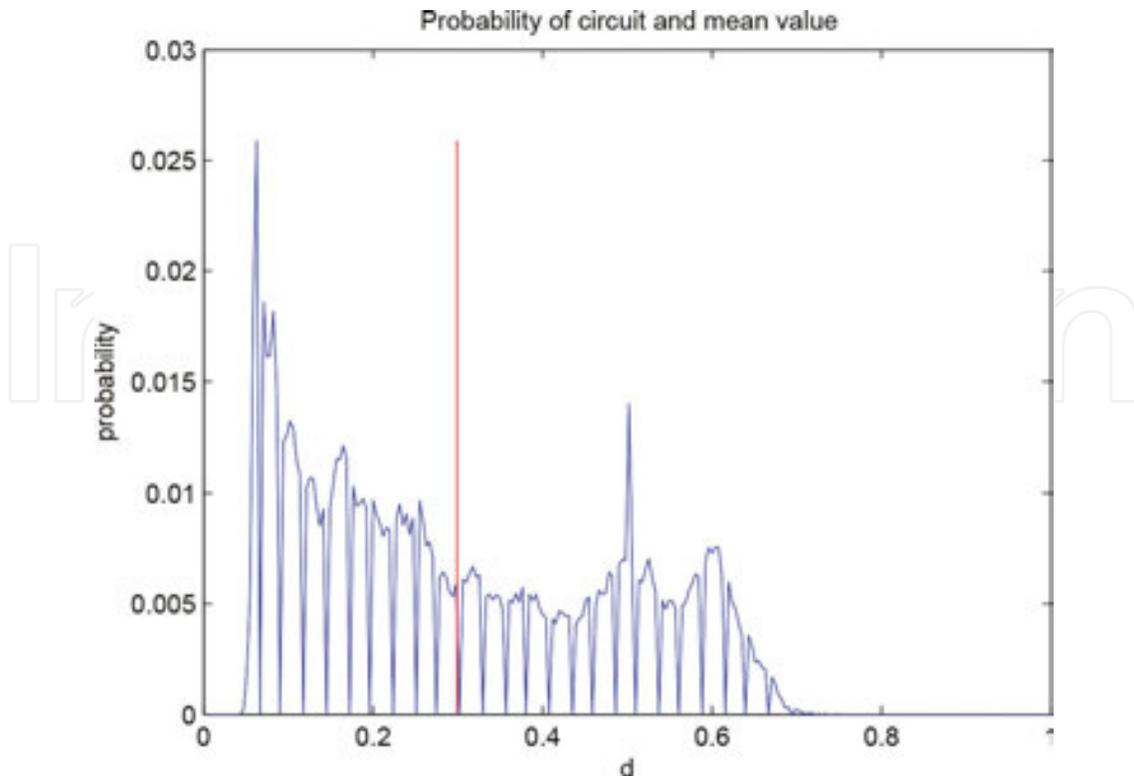


Figure 3. Mean value.

There are two other parameters for the distribution location, median, and mode. The median of the distribution is a value separating the higher half of plot from the lower half, it is a d value. The mode is the value that appears most often in a set of data, the one with highest probability (the d , where is highest H). MATLAB has implemented functions:

```
%compute median from the image;
Me = median ( reshape ( Im,size (Im,1)*size(Im,2),1 ) )
%compute mode from the image;
Mo = mode ( reshape ( Im,size (Im,1)*size (Im,2),1 ) )
figure, plot (d, H);
xlabel ('d');
ylabel ('probability');
title ('Probability of circuit with median and mode values');
%add median value to the probability plot;
hold on, plot ([Me, Me], [0, max(H)], '-.*r');
%add mode value to the probability plot;
hold on, plot ([Mo, Mo], [0, max (H)], '.*g');
```

When plotted, median is dash-dotted red and mode-dotted green. The median often serves instead of mean for the distributions that are not Gaussian. The mode expresses the most frequent value in the distributions that has only one such peak, and thus they are unimodal (Figures 8 and 9).

The second parameter of the distribution is the scaling parameter. It describes how far the other d values are from the location parameter μ . The second central moment estimates the variance σ^2 , measures how far the d values are spread out (dispersed). An equivalent measure is the square root of the variance, called the standard deviation σ . Standard deviation thus measure dispersion of the d values.

$$\sigma^2 = \sum_k H_k * (d_k - \mu)^2 \quad (5)$$

```
%compute variance from the normalized histogram;
sigma2 = sum (H.*(d-mu). ^2)
%compute standard deviation;
sigma = sqrt (sigma2)
or
%compute variance from the image;
sigma2 = var ( reshape ( Im,size (Im,1)*size (Im,2),1 ) )
%compute standard deviation;
sigma = std ( reshape ( Im,size (Im,1)*size (Im,2),1 ) )
```

In case, that the dispersion is skewed, thus has different dispersions on left and right side of the plot (from the point of view of the location parameter), it is recommended to use the Inter Quartile Range *IQR* as robust measure of scale. Usage of the *IQR* also removes the affects of the outliers to the distribution dispersion.

```
%compute inter quartile range;
Q = iqr (reshape (Im,size (Im,1) * size (Im,2),1 ) )
```

The value of the Inter Quartile Range *IQR* is usually bigger than the standard deviation σ .⁵

However, the basic statistical parameters do not cover the distributions that have more than one mode (multimodal), and also cannot describe the negative exponential distributions without location parameter. In that case, we are using different measure of the distribution, the entropy S . Entropy is a measure of unpredictability of information content:

⁵ Approximately 1.349 times for Gaussian distribution

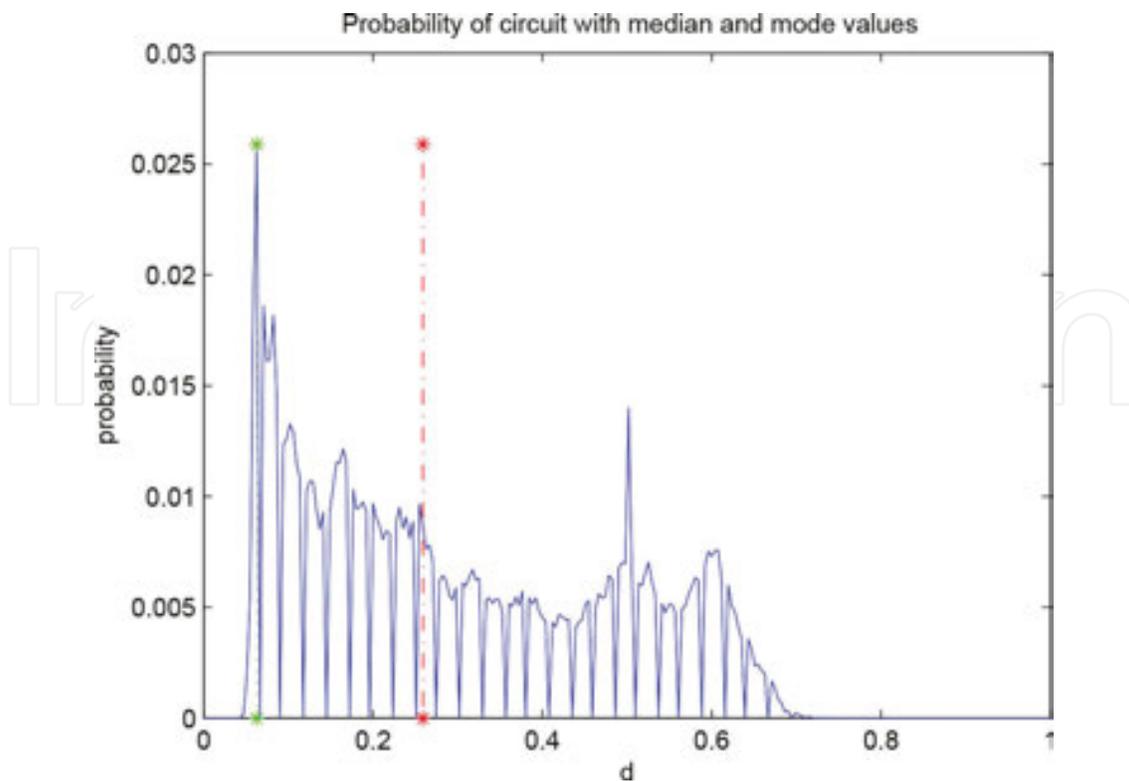


Figure 4. Media and mode values.

```
%compute entropy of the image;
```

```
S = entropy (Im)
```

or

```
%find where normalized histogram equals zero;
```

```
f = find (H==0);
```

```
%exclude zero values from computation;
```

```
Hx = H;
```

```
Hx(f) = [];
```

```
%compute entropy from the normalized histogram without zero values;
```

```
S = -sum (Hx.*log2 (Hx))
```

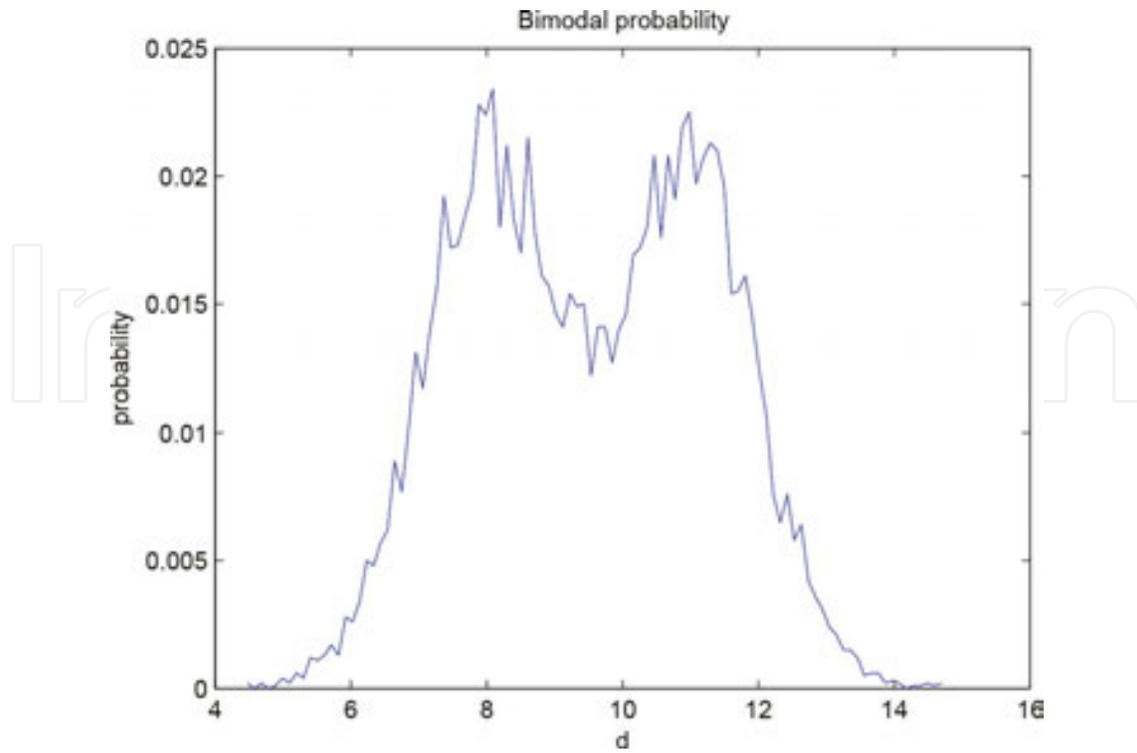


Figure 5. Bimodal distribution.

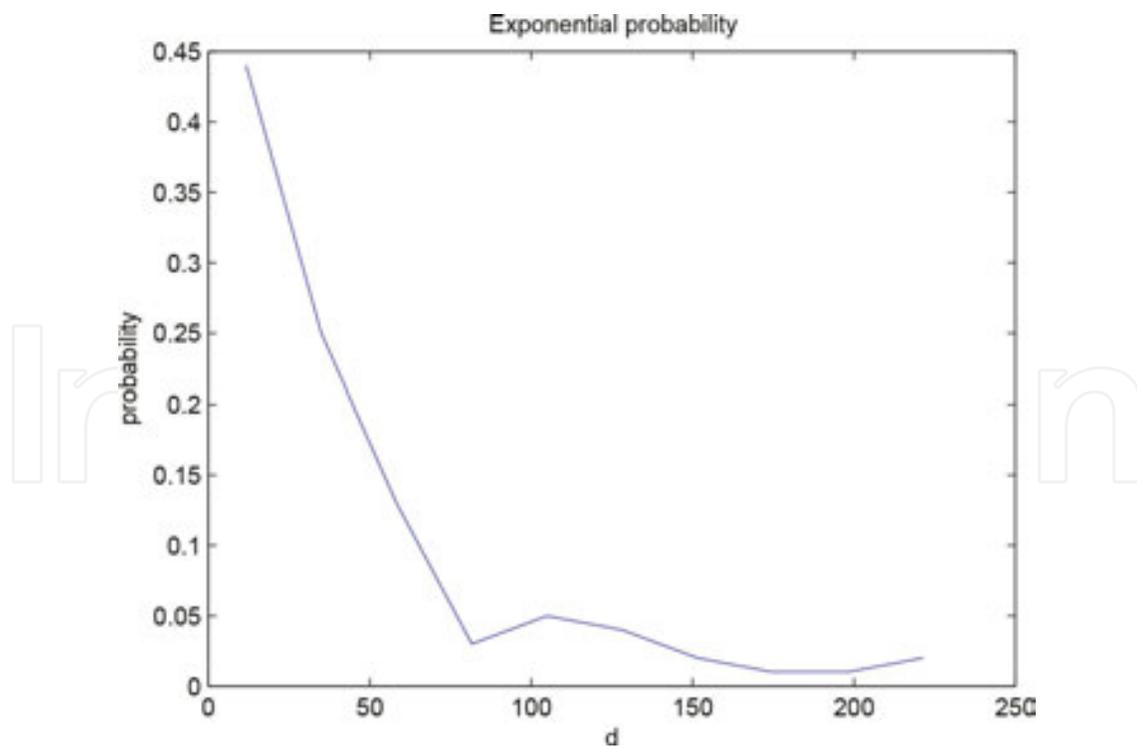


Figure 6. Exponential distribution.

Circuit, entropy $S = 6.9439$

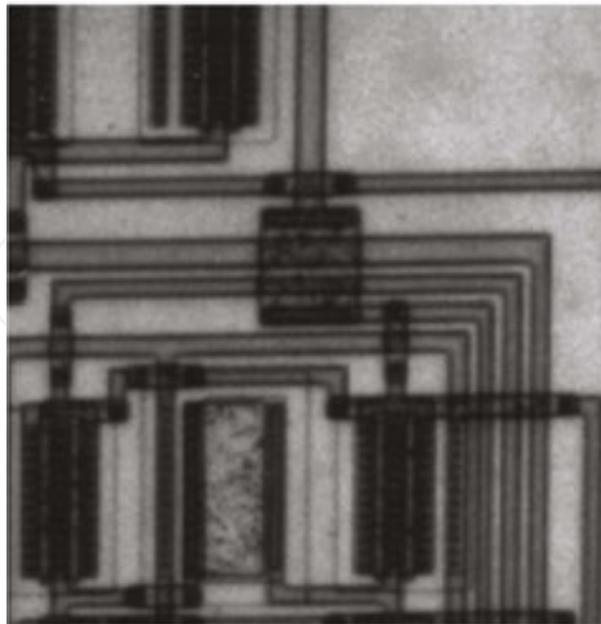


Figure 7. Entropy of the circuit.

Histogram of circuit, entropy $S = 6.9439$

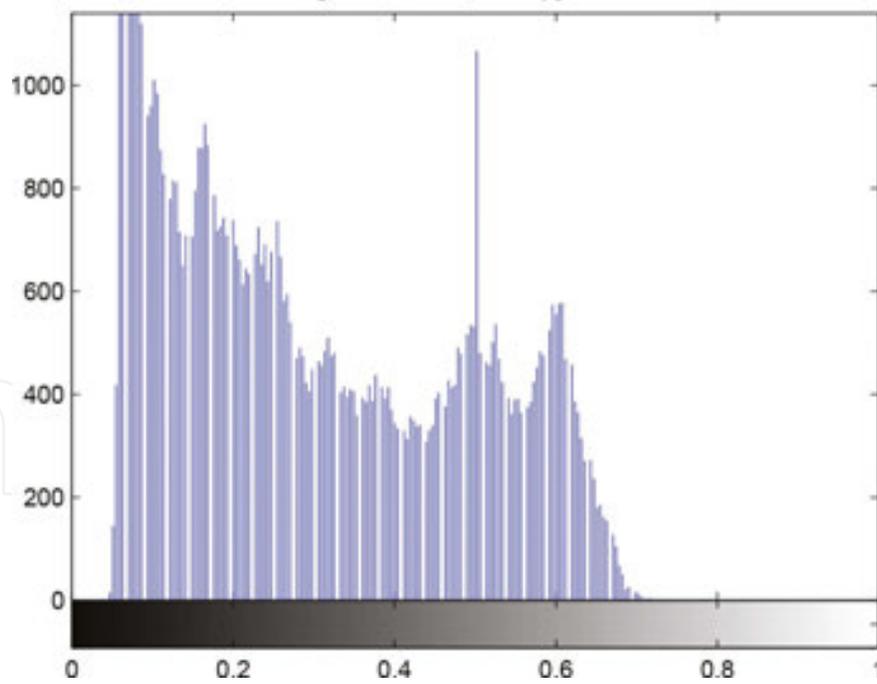


Figure 8. Entropy and histogram of the circuit.

Gaussian noise, entropy $S = 7.6195$

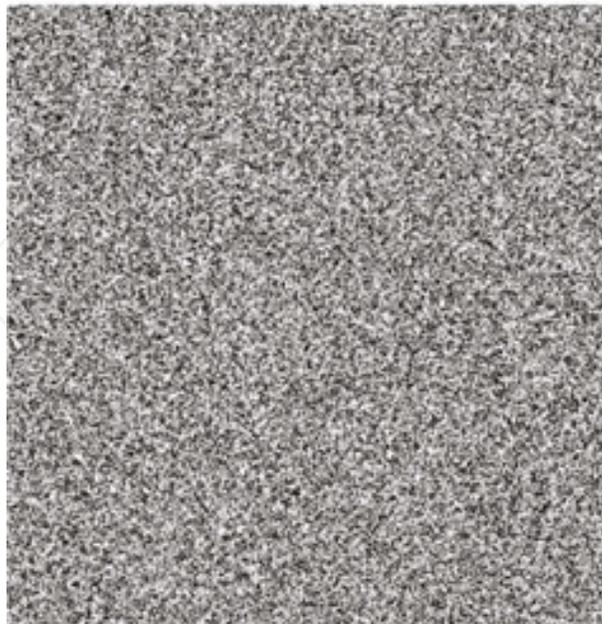


Figure 9. Entropy of the Gaussian distributed intensities, $\mu = 0.5$, $\sigma = 0.1$.

Histogram of Gaussian noise, entropy $S = 7.6195$

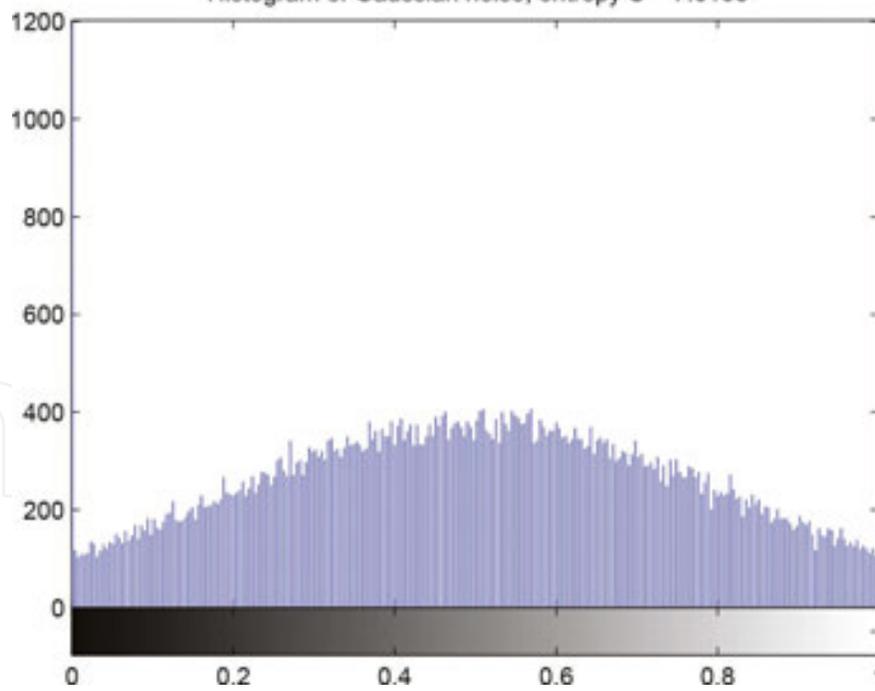


Figure 10. Entropy and histogram of the Gaussian distributed intensities, $\mu = 0.5$, $\sigma = 0.1$.

Cells, entropy $S = 4.6024$



Figure 11. Entropy of the cells.

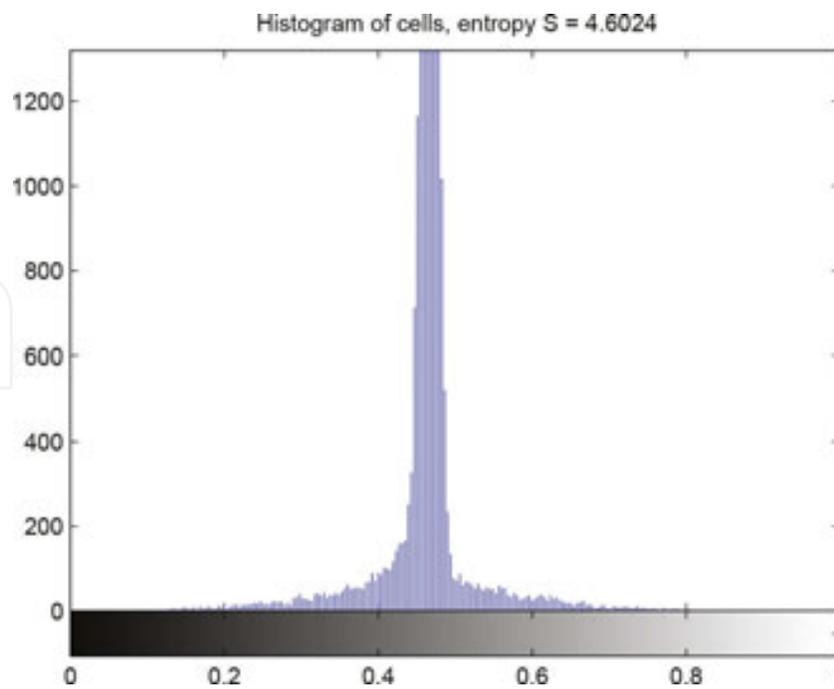


Figure 12. Entropy and histogram of the cells.

Unique intensity value, entropy $S = 0$



Figure 13. Entropy of the unique intensity value for all pixels $d=0.5$.

Histogram of unique intensity value, entropy $S = 0$

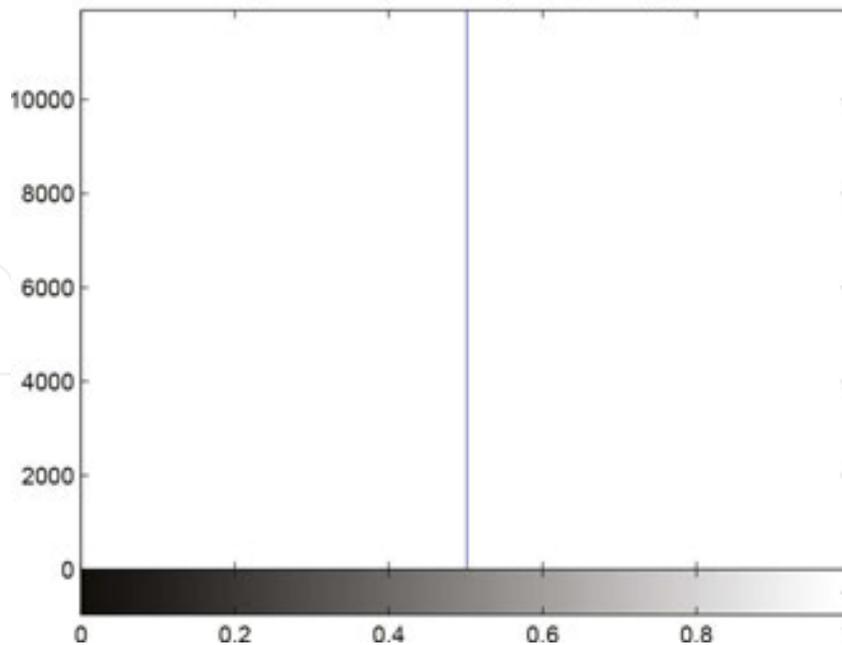


Figure 14. Entropy and histograms of the unique intensity value for all pixels $d=0.5$.

There will be completely different values of entropy S for images with different distributions/histograms.

```
%show histogram and entropy of circuit image;
figure, imshow (Im);
title (['Circuit, entropy S = ', num2str(S) ] );
figure, imhist (Im);
title (['Histogram of circuit, entropy S = ', num2str(S) ] );
%show histogram and entropy of Gaussian noise;
J = imnoise (zeros (size (Im)), 'Gaussian',.5, .1);
SJ = entropy(J)
figure, imshow(J);
title (['Gaussian noise, entropy S = ',num2str (SJ)]);
figure, imhist(J);
title (['Histogram of Gaussian noise, entropy S = ',num2str(SJ) ] );
%show histogram and entropy of cell image;
C = imread ('cell.tif');
C = double (C) / 255;
SC = entropy (C)
figure, imshow (C);
title (['Cells, entropy S = ', num2str (SC) ] );
figure, imhist (C);
title (['Histogram of cells, entropy S = ',num2str (SC) ] );
%show histogram and entropy of unique value;
U = ones (size (Im))*5;
SU = entropy (U)
figure, imshow (U);
title (['Unique intensity value, entropy S = ', num2str(SU) ] );
figure, imhist(U);
title (['Histogram of unique intensity value, entropy S = ',num2str(SU) ] );
```

Entropy is a number that somehow characterize the distribution:

Distribution	S
Circuit	6.9439
Gaussian	7.6278
Cells	4.6024
Unique	0
Bimodal	4.2245
Exponential	1.5928

3. Entropy

Let start with a reminder of the form of Shannon entropy from information theory with respect to the image analysis terminology. Any given normalized discrete probability distribution $H * h_1, h_2, \dots, h_D *$ fulfills the condition:

$$\begin{aligned} h_d &\geq 0 \\ \sum_{d=1}^D h_d &= 1. \end{aligned} \tag{6}$$

Usually, in an intensity image, there exists an approximation of probability distribution given by the normalized histogram function $H(d)$. H is an intensity function that shows the count of the pixels $\Phi(i, j)$ with intensity equals to d independent on the image position (i, j) [3,4]. The histogram is normalized by the number of pixels to fulfill the conditions.⁶

More conditions are assumed when measuring the information. Information must be additive for two independent events a, b :

$$I(ab) = I(a) + I(b). \tag{7}$$

The information itself should be dependent only on the probability distribution or normalized histogram function in our cases. Equation 7 describes the conditions referred to. This equation is the well-known modified Cauchy's functional equation with unique solution $I(h) = -\kappa \times \log_2(h)$. In statistical thermodynamic theory, the constant κ refers to the Boltzman constant [5]. In the Hartley measure of information, κ equals one [6,7]. Let us focus on Hartley measure. If different amounts of information occur with different probabilities, the total

⁶ The 0th central moment again.

amount of information is the average of the individual information, weighted by the probabilities of their individual occurrences [7,8]. Therefore, the total amount of information is:

$$\sum_d (h_d I_d), \quad (8)$$

which leads us to the definition of Shannon entropy as a measure of information:

$$S = -\sum_d h_d \log_2(h_d). \quad (9)$$

Thus, entropy is the sum of the individual information weighted by the probabilities of their occurrences.

In image analysis, the unknown probability distribution function of intensity values is approximated via histogram function $H(d)$: The histogram $H(d)$ has to be normalized to the total amount of pixels [9,10]. Shannon entropy allows information content of the whole image or just from the selected part of the image to be measured (**Figures 10 and 11**).

The entropy implemented in MATLAB function

`S = entropy(Im)`

is Shannon entropy.

4. Entropy filtration

Entropy allows all the information content of the entire image to be measured. However, when we change the number of pixels in the histogram computation, we obtain partial information content that is strictly dependent on the area entering the computation (**Figures 12 and 13**).

Entropy filtering is based on the replacement of pixel values in the image by values of entropy. Entropy is computed in a specified area, usually from the pixel's n -by- n symmetric neighborhood in the input image [4,11]. The shape of the neighborhood should be also defined by the users. The computed entropy is

$$F_{(i,j)} = -\sum h_{d,(se)} \log_2(h_{d,(se)}), \quad (10)$$

where $se(i,j)$ is the pixel's $\Phi(i,j)$ neighborhood.

Entropy filtering of circuit, $se = true(9)$

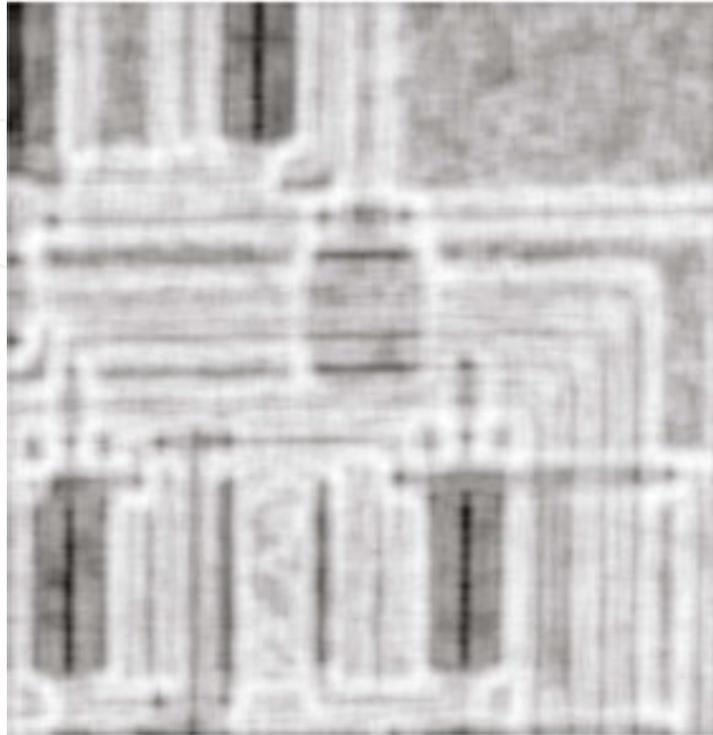


Figure 15. Entropy filtering of circuit image with $se = true(9)$.

Entropy filtering of circuit, $se = true(41)$



Figure 16. Entropy filtering of circuit image with $se = true(41)$.

Entropy filtering of circuit, se = true(91)



Figure 17. Entropy filtering of circuit image with $se = true(91)$.

It is clear that the output image (as computed by entropy filtration) is strongly dependent on the area selected. For small se , the local disturbances will be given sufficient weight, and the output image will be too noisy. On the other hand, too large an se value will not preserve details and the output image will be blurred. Therefore, the key question in the filtration method is how to select a suitable neighborhood. se selection is always a compromise between a noisy or blurry image. Of course, filtration can be very useful for decreasing the area and thus allowing further analysis (**Figures 14** and **15**).

```
%compute entropy filtering with small structure element;
F = entropyfilt (Im);
figure, imshow (F,[]);
title ('Entropy filtering of circuit, se = true(9)');
%compute entropy filtering with middle structure element;
F = entropyfilt (Im,true (41));
figure, imshow (F,[]);
title ('Entropy filtering of circuit, se = true(41)');
%compute entropy filtering with large structure element;
F = entropyfilt (Im,true(91));
figure, imshow (F,[]);
title ('Entropy filtering of circuit, se = true(91)');
```

5. Entropy thresholding and segmentation

Thresholding is a time cheap method searching for point in the intensity histogram H for separating image into the objects related to the real objects. It takes from the image parts that corresponding to the threshold parameter(s). Automatic threshold selection using the entropy is based on the maximization of entropy segmentation. The histogram function $H(d)$ is separated into two parts, A and B , iteratively in d . For both parts, the Shannon's entropies are computed

$$S_A = -\sum_{w=1}^{w=k} h_w \log_2(h_w) \quad (11)$$

$$S_B = -\sum_{w=k}^{w=D} h_w \log_2(h_w), k = d_2 : d_{max-1}. \quad (12)$$

Then, the entropy of part A and B (taken together) is computed as

$$S_V = -\log_2(H_A) - \log_2(H_B) - \frac{S(A)}{H_A} - \frac{S(B)}{H_B}. \quad (13)$$

A threshold value is set for d , where SV_k is maximized [12,13]. This method uses the global histogram function; therefore, it is not sensitive to the random noise contribution and successfully removes the noise. However, the use of thresholds also ignores local changes in the background, illumination, and non-uniformity. For images with different conditions within the scene, thresholds generally produce loses and artifacts. The use of thresholds without any previous preprocessing, for example,, light normalization, is applicable only with objects that are well separable from the background. Automatic segmentation techniques [3,4,12,14,15] are very powerful tools under easily-separable conditions (**Figures 16 and 17**).

```
HA = zeros(size(H)); %empty lower histogram;
```

```
HB = zeros(size(H)); %empty upper histogram;
```

```
%empty cumulative distribution function;
```

```
C = zeros(size(H));
```

```
%cumulative distribution function;
```

```
C(1) = H(1);
```

```
for k = 2:length(H),
```

```
    C(k) = C(k-1) + H(k);
```

```
end;
```

```

C = double(C);
%cycle through intensity levels;
for k = 1:length(H),
if C(k) > 0, %only for positive cumulation;
for w = 1:k, %from beginning till now;
if H(w) > 0, %only for positive histogram
%compute the lower histogram value
HA(k) = HA(k) - ( H(w)/C(k)) * log2(H(w)/C(k) );
end; %endif;
end; %endfor;
end; %endif;
if ( 1-C(k) ) > 0, %only for positive cumulation residuals;
for w = k + 1:length(H); %from now till end;
if H(w) > 0, %only for positive histogram
%compute the lower histogram value
HB(k) = HB(k) - ( H(w)/(1-C(k))) * log2(H(w)/(1-C(k) ) );
end; %endif;
end; %endfor
end; %endif
end; %endfor
%locate the maxima for joined histograms
[co, kde] = max(HA+HB);
%selet threshold
Th = d( kde-1 )
%segment image
II = im2bw(Im, Th);
figure, imshow(II);
title(['Entropy segmentation of circuit, Th = ', num2str(Th)]);

```

The value d where the entropy SV is maximized represents the threshold for segmentation of the image (**Figure 18**).

Otsu segmentation of circuit, Th = 0.33333



Figure 19. Segmentation of circuit image by Otsu.

Entropy segmentation of circuit, Th = 0.34118



Figure 18. Segmentation of circuit image by entropy.

5.1. Grayscale thresholding

The entropy segmentation gives similar results with the Otsu thresholding. Otsu gray level thresholding is a nonparametric method of automatic threshold selection for image segmentation also from the normalized intensity histogram $H(d)$. For separating histogram into two classes, the between class variance is maximized:

```
%cycle through the histogram;
for T=2:length(H)-1,
w(1) = sum( H(1:T) ); %probability of first class
u(1) = sum( H(1:T) .* d(1:T) ); %class mean
%protection against zero;
if w(1) == 0,
u(1) = 0;
else
%class mean recomputation;
u(1) = u(1)/w(1);
end;
w(2) = sum( H( (T+1):end) ); %probability of second class
u(2) = sum( H( (T+1):end) .* d( (T+1):end) ); %class mean
%protection against zero;
if w(2) == 0,
u(2) = 0;
else
%class mean recomputation;
u(2) = u(2)/w(2);
end;
%between class variance;
ut = w*u';
sigmaB(T) = w(1)*(u(1)-ut)^2 + w(2)*(u(2)-ut)^2;
end;
%find maximal between class variance
[e,r] = max(sigmaB);
```

```
TTh(1) = d(r); %set Threshold
or
TTh = graythresh(Im); %compute threshold;
IO = im2bw(Im, TTh); %segment image;
figure, imshow(IO);
title(['Otsu segmentation of circuit, Th = ', num2str(TTh)]);
```

6. Point Information Gain

The most interesting is the point information gain (PIG) which asks the question: How important is one pixel for the whole image or for the selected part? In other words, is the occurrence of the value of one single pixel a surprise? It is predictable that for value of background pixels it will not carry a lot of information, if we discard one of them. On the other hand, the objects, especially if they are complicated in structure, will increase the entropy on their position. Shannon equation evaluate total amount of the information entropy from the whole histogram. Let evaluate the normalized image histogram $H(d)$ and compute the Shannon information entropy S :

$$S = -\sum_d H_d \log_2(H_d), \quad (14)$$

To investigate the contribution of one single pixel with intensity value v to the total entropy, we need to evaluate the second histogram $G(d)$ which is created without this investigated pixel:

$$\begin{aligned} g(i, j, d) &= q(i, j, d), \text{ if } d \neq v; \\ g(i, j, d) &= q(i, j, d) - 1, \text{ if } d = v. \end{aligned} \quad (15)$$

This time we discard the value v of the center investigated $\Phi(i, j)$ from the computation, but only once.

One single pixel of intensity value d will only decrease the histogram value $g(d)$ on its intensity position d . Then, the histogram is again normalized. The probability of intensity value d is slightly lower than the probability $H(d)$ of the primary normalized histogram (with all pixels). The other probabilities $g(d)$, where d is not the value of investigated pixel, are slightly higher than the probability $H(d)$ of the primary normalized histogram (with all pixels). Then, in the second computation of entropy E , computed from the modified normalized histogram $G(r)$:

$$E = -\sum_d g_d \log_2(g_d), \quad (16)$$

the individual information $\log_2(g(d))$ as well as their weights $g(d)$ differs according to the computation of whole entropy S . Therefore, we obtained two different entropy values S and E . Entropy S represents the whole measure of information in original image. Entropy E represents the measure of information in the image without the investigated pixel. The difference PIG [16]:

$$PIG = S - E, \quad (17)$$

refers to the difference between the entropy of the two histograms, and therefore also difference between the entropy of the two images (the first one with contains our investigated pixel $\Phi(i, j)$ and the second one without this investigated pixel). Recall that the both histograms H and G were normalized, and therefore, any difference in the number of pixels in the images is immaterial. Difference PIG represents either the entropy contribution of pixel $\Phi(i, j)$ or the contribution of the value of the pixel $\Phi(i, j)$ to the information content of the whole image. The transformation of each image pixel $\Phi(i, j)$ value to its contribution to the whole image via equation 17 represents the measure of the information carried by that pixel, the Point Information Gain (PIG). Repeated computation 17 for every single pixel of the image transforms the original image into the entropy map: the image that shows contribution of every pixel to the whole information content of the image (Figures 19 and 20).

Point Information Gain of the cells.

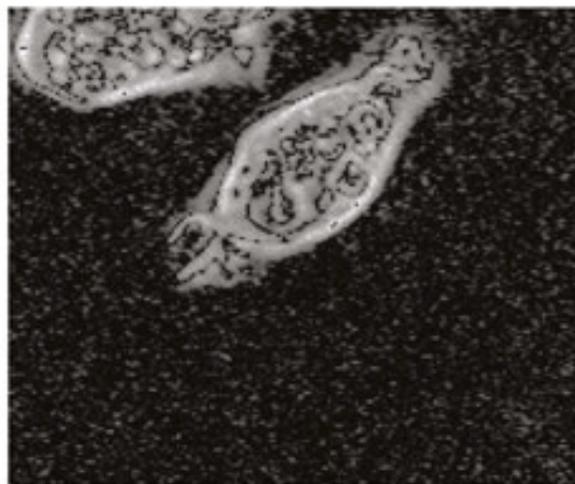


Figure 20. Point information gain of grayscale image cells.

It is predictable that the values of background pixels will not carry a lot of information, even if we discard one of them. On the other, the objects, especially if they are complicated in structure, will increase the entropy in their immediate area. According to the information

theory, the object occurrence produces a bigger surprise than does background occurrence, and the *PIG* quantifies this effect. For this reason, the details in the image are preserved: they are the surprise. For the same reason, random noise is removed: We always know it is presented, and no surprise occurs (**Figures 21 and 22**).

Point Information Gain of the cameraman.



Figure 21. Point Information Gain of grayscale image cameraman.

Point Information Gain of the circuit.

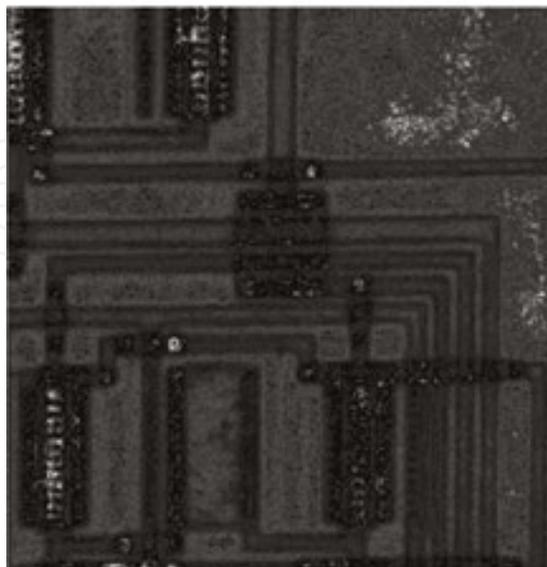


Figure 22. Point Information Gain of grayscale image circuit.

```

%load images(s)
C = imread('cell.tif');
%C = imread('cameraman.tif');
%C = imread('circuit.tif');
C = double(C)/255;
S = entropy(C); %compute entropy
%compute average probability of one pixel
pomo = 1/numel(C);
[H,d] = imhist(C); %compute histogram
H = H./sum(H); %normalize histogram
IE = zeros(size(C)); %empty result image;
%cycle through intensity levels;
for k=1:length(H);
%precompute second histograms;
G = H;
%remove pixel contribution;
if G(k)>=pomo,
G(k) = G(k) - pomo;
end;
%protection against zero;
f = find(G==0);
G(f) = [];
G = G./sum(G); %renormalization;
%entropy without pixel;
E(k) = -sum(G.*log2(G));
%point information gain;
PIG(k) = S-E(k);
%assign pig to pixels;
f = find(C==k/255);
IE(f) = PIG(k);

```

```
end;
figure, imshow(IE,[]);
title('Point Information Gain of the cells.');
```

%title('Point Information Gain of the cameraman.');

%title('Point Information Gain of the circuit.');

7. Conclusion and discussion

For those, who are interested in the entropy processing, the things are little bit more complicated.

PIG approach is dependent only on pixel $\Phi(i, j)$, there is no information about pixel's position. Therefore, the area *se* of the histogram function computation could include not the whole image, but only some selected area around the investigated pixel. The *se* could be the whole row and whole column in which the pixel is located. Difference $PIG = S - E$ in this case refers to the difference between the information content of the two crosses. Difference *PIG* represents either the entropy contribution of pixel $\Phi(i, j)$ or the contribution of the value of pixel $\Phi(i, j)$ to the cross. Even more derivation from the original *PIG* algorithm were developed recently [17–19].

There also exist different entropies, not only Shannon, namely Tsallis-Havrda-Charvát and Rényi definitions at least. The Rényi entropy:

$$R_{\alpha} = \frac{1}{1-\alpha} \log_2 \left(\sum_d h_d^{\alpha} \right), \quad (18)$$

is the generalization of the Shannon entropy. For the $\alpha=0$, the Rényi entropy equals Shannon ($R_0=S$).

The evaluation of entropy has heavy computational burden; therefore, it is recommended to use parallelization on GPU. For the processing of the color images, it is usual to tread each color channel independently like a grayscale image.

Overall, the entropy is a representative parameter of the image and there is still a lot of potential in its usage for processing and analysis.

The code presented in this chapter could be downloaded at: <https://www.mathworks.com/matlabcentral/fileexchange/55493-information-entropy>.

Acknowledgements

This work was supported by the Ministry of Education, Youth, and Sports of the Czech Republic - projects "CENAKVA" (No. CZ.1.05/2.1.00/01.0024) and "CENAKVA II" (No. LO1205 under the NPU I program).

Author details

Jan Urban

Address all correspondence to: urbanj@frov.jcu.cz

Laboratory of Signal and Image Processing, Institute of Complex Systems, South Bohemian Research Center of Aquaculture and Biodiversity of Hydrosystems, FFPW, USB, Zámek, Nové Hradky, Czech Republic

References

- [1] Katajama, M., Orešič, M. Data processing for mass spectrometry-based metabolomics. *Journal of Chromatography A*. 1158:318–328, 2007.
- [2] Pearson, K. Contributions to the mathematical theory of evolution. II. Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 186:343–414, 1985.
- [3] Sonka M., Hlavac V., Boyle R. *Image processing, analysis and machine vision*. Brooks/Cole Publishing Company, 1999.
- [4] Gonzales R. C., Woods R. E. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [5] Boublik T. *Statistical thermodynamic*. Academia, 1996.
- [6] Hatley J. V. *Bell System Technical Journal*. 7:535, 1928
- [7] Jizba P., Arimitsu T. The world according to Renyi: thermodynamics of multifractal systems. *Annals of Physics*. 312:17–59, 2004.
- [8] Shannon C. E. A mathematical theory of communication. *Bell System Technical Journal*. 27:379–423 and 623–656, 1948.
- [9] Demirkaya O., Asyali M. H., Sahoo P. K. *Image processing with MATLAB: Applications in medicine and biology*. CRC Press, 2009.
- [10] Nixon M., Aguado A. *Feature extraction & image processing*. Academic Press, 2002.

- [11] Moddemeijer R. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*. 16(3):233–246, 1989.
- [12] Pun T. A new method for grey level thresholding using the entropy of the histogram. *Signal Processing*. 2:223–237, 1980.
- [13] Tzvetkov P., Petrov G., Iliev P. Multidimensional dynamic scene analysis for video security applications. *IEEE Computer Science' 2006, Istanbul*.
- [14] Beucher S. Applications of mathematical morphology in material sciences: a review of recent developments. *International Metallography Conference*, pp. 41–46, 1995.
- [15] Otsu N. A Threshold Selection Method from Gray-Level Histogram. *IEEE Transactions on Systems, Man, and Cybernetics* 9:62–66, 1979.
- [16] Urban J., Vanek J., Stys D. Preprocessing of microscopy images via Shannon's entropy. In *Proceedings of Pattern Recognition and Information Processing*: pp.183–187, Minsk, Belarus, ISBN 978-985-476-704-8, 2009.
- [17] Rychtarikova R., Nahlik T., Smaha R., Urban J., Stys D. Jr., Cisar P., Stys D. Multifractality in imaging: application of information entropy for observation of inner dynamics inside of an unlabeled living cell in bright-field microscopy. In *ISCS14*, Sanayei et al. (eds.), Springer, pp. 261–267, 2015.
- [18] Štys D., Urban J., Vane-k J., Cisar- P.. Analysis of biological time-lapse microscopic experiment from the point of view of the information theory, *Micron*, S0968-4328(10)00026-0, 2010.
- [19] Štys D., Vane-k J., Náhlík T., Urban J., Cisar- P.. The cell monolayer trajectory from the system state point of view. *Molecular Biosystems*, 7:2824–2833, 2011.

