Rank-based Transformation Algorithm (RBT) for Image Contrast Adjustment

Cheng-Hui Chen and Torbjörn E. M. Nordling

Affiliation not available

October 31, 2023

Abstract

Performing proper image contrast adjustment without information loss is an art. The default settings are often inappropriate for the image in question rendering a contrast adjustment depending on trial and error. We propose a simple method, rank-based transformation (RBT), for image contrast adjustment that requires no prior knowledge. This makes RBT an ideal first tool to apply for underexposed images. The RBT algorithm normalizes and equalizes all the intensity differences of the image over the full intensity range of the image data type, and thus assigning equal weight to all gradients. Even the state-of-the-art AI tool Cellpose visually benefits from RBT preprocessing. Our comparison of histogram normalization methods demonstrates the ability of RBT to bring out image features.
Rank-based Transformation (RBT) Algorithm for Image Contrast Adjustment

Cheng-Hui Chen, Torbjörn E. M. Nordling Member, IEEE,

Abstract—Performing proper image contrast adjustment without information loss is an art. The default settings are often inappropriate for the image in question rendering a contrast adjustment depending on trial and error. We propose a simple method, rank-based transformation (RBT), for image contrast adjustment that requires no prior knowledge. This makes RBT an ideal first tool to apply for underexposed images. The RBT algorithm normalizes and equalizes all the intensity differences of the image over the full intensity range of the image data type, and thus assigning equal weight to all gradients. Even the state-of-the-art tool Cellpose visually benefits from RBT preprocessing. Our comparison of histogram normalization methods demonstrates the ability of RBT to bring out image features.

Index Terms—keywords: image processing, rank-based transformation (RBT), contrast adjustment, histogram equalization, histogram expansion, histogram normalization.

I. INTRODUCTION

Image processing plays a crucial role in a variety of research fields, including biology [1], microbiology [2], medicine [3]–[5], neurology [6], [7], computer vision [8], and machine learning [9], [10]. Despite advances in machine learning techniques for image processing and recognition, images preprocessing is still crucial [11]. One important aspect of preprocessing is contrast adjustment. This is necessary when an image is recorded under inappropriate exposure conditions. The goal of image processing is to enhance relevant information while reducing noise and background interference. A common method of image enhancement is to modify the histogram of the image, which is a representation of the distribution of pixels over the tonal range of the image.

The histogram is divided into intervals called “value bins”, and the intensity values that fall into each bin are calculated. For example, underexposed or overexposed images tend to have their pixel intensity value bins clustered on one side of their tonal range, resulting in poor contrast in the image [12]. On the other hand, if the histograms are distributed uniformly over the possible ranges of intensity level (depending on the data type of the image), the images tend to present high contrast and show more gray-level details [13].

A myriad of terms are used to describe different aspects of image enhancement, sometimes with seemingly conflicting definitions. Thus we provide an overview in a hierarchical structure (Fig. 1) and our usage of the terms. Image enhancement is a broad term referring to the process of improving the quality or visual appearance of an image, which can be achieved through various techniques like contrast enhancement, noise reduction, and sharpening [14]. Image adjustment refers to the process of altering specific aspects or parameters of an image, such as brightness, contrast, color balance, and gamma correction, to achieve desired output, and can thus be considered a subset of image enhancement [15]. Within image enhancement, contrast adjustment refer to the process of modifying the contrast of an image to make it more visually appealing or to reveal hidden details, while contrast enhancement focuses on improving the contrast between different regions or objects in an image. Histogram modification or adjustment refer to the process of altering the histogram of an image to adjust its characteristics, such as contrast or brightness, while histogram enhancement emphasize the improvement of the image’s visual quality or appearance through histogram adjustments [16]. Histogram expansion refers to the process of stretching the original dynamic range (DR) of an image to a larger DR, typically the full available DR of the data type, in order to enhance contrast by increasing the intervals between histogram bins. Histogram equalization (HE) and histogram normalization (HN) are two distinct contrast enhancement techniques that involve modifying the histogram to optimize the distribution of intensities [17]. HE aims to expand and flatten the histogram of an image globally, uniformly distributing intensity over the available DR. [18] HE can be classified into global histogram equalization (GHE) and local histogram equalization (LHE), where the equalization in the later case is done upon the local regions of the image. In contrast, HN scales the pixel intensities of an image to a standard range, typically [0, 1] or [0, 255]. HN methods can be classified as linear and nonlinear ones.

Image contrast adjustment is still an active research topic. We discuss recent work in relation to well-known methods. The contrast adjustment with GHE involves normalizing the intensity distribution using its cumulative distribution function, resulting in a uniform distribution of intensity in the output image [19]. However, the global equalization is not suitable for images with non-uniform distributed intensity over all regions. The drawbacks including the over-enhancing of local region, increasing of noise level, losing of information and details, causing wash-out effect on some approximate-homogeneous regions [15], [19], [20]. LHE intends to overcome these drawbacks.

Adaptive histogram equalization (AHE) is a LHE method that works on small regions of the image instead of the entire image. This allows for better preservation of the image details,
but can also result in the increase of noise in low contrast regions. Various variations of AHE have been proposed to limit the enhancement of local noise, such as contrast limited adaptive histogram equalization (CLAHE), brightness preserving dynamic histogram equalization (BPDHE), dualistic sub-image histogram equalization (DSIHE), and center-excluded histogram equalization (CEE) [17].

The CLAHE is a modified AHE method that remaps each pixel based on its local (neighborhood) grayscale distribution [23], [24]. CLAHE works by dividing the image into small, overlapping regions called “tiles”, applying HE to each tile, and limiting the contrast amplification in each tile by clipping the histogram of each region at a given threshold. The clipped portion is added back to the region below the threshold to maintain the same total area of the histogram [17]. CLAHE reduces the over-amplification of noise in homogeneous areas and improves the contrast of textured regions [25], [26]. However, CLAHE may cause blocking effects due to abrupt changes in the neighboring histograms at the boundaries of the regions. Also, the choice of the clipping threshold affects the performance, as high values can reduce the contrast enhancement while low values can introduce artifacts [5], [24].

Both GHE and LHE can be combined with other histogram modification (HM) methods, such as gamma correction [15], [23], [27] and fuzzy clustering [28], [29], to improve their performance. These methods require users to identify and specify proper parameters to achieve desired quality of output images. The process of identifying the proper parameters are usually time consuming and tedious. Some optimization methods such as genetic algorithm (GA) [30], particle swarm optimization (PSO) [31], fuzzy-based improved particle swarm optimization (FIPSO) [32] are then combined to help identifying proper parameters for image adjustment.

Histogram normalization is the process of remapping or rescaling the intensity value of each pixel in the image to an expanded DR using either linear or nonlinear transformation functions. The linear transformation function is denoted as

\[
y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}(y_{\max} - y_{\min}) + y_{\min}. \tag{1}
\]

The intensity of a pixel on the input image is denoted by \(x\), with the range \([x_{\min}, x_{\max}]\), and the output intensity is \(y\), with the range \([y_{\min}, y_{\max}]\) to which the histogram is expanded. The alternative name of this method is linear contrast expansion, and it can also be applied to color images by operating on the brightness [12].

Multiplicative scaling involves directly multiplying the original intensity values by a constant \(c\) greater than zero to either compress (if \(c < 1\)) or stretch (if \(c > 1\)) the DR of the image [18]. The general multiplicative scaling function is denoted as

\[
y = c \times x. \tag{2}
\]

If \(c\) is equal to \(y_{\max}/x_{\max}\), and \(x_{\min}\) is not zero, then the lower part of the available DR of the data type is not utilized. For an underexposed image, this method expands the distances between the intensity values, and thus adjusts the contrast [18]. In (1) and (2), the parameters to be specified before the transformation are \([y_{\min}, y_{\max}]\) and \(c\).

![Fig. 1. Categories of well-known histogram modification methods.](image)

There are two nonlinear normalizations that are mentioned in textbooks, the logarithmic normalization and the power-law normalization. The general form of the logarithmic normalization is denoted as

\[
y = c \times \log_a(1 + x). \tag{3}
\]

To ensure \(y_{\max}\) within the target range, and to decrease the numerical value of \(c\), we rearrange (3) as

\[
y = \frac{y_{\max}}{\ln(2)} \times \ln \left(1 + \frac{x - x_{\min}}{x_{\max} - x_{\min}}\right) + y_{\min}. \tag{4}
\]

When the constant \(c\) is larger, the output image will be brighter. To utilize the full DR of \(y\), the maximum \(c\) is specified to \(y_{\max}/\log_a(2)\). The 1 in (3) and (4) is to avoid \(\log_a(0)\), i.e. infinity [18], [33]

The general form of power-law normalization is written as

\[
y = c \times x^\gamma. \tag{5}
\]

To fix \(y\) to the target range \([y_{\min}, y_{\max}]\), we rearrange (5) as

\[
y = (y_{\max} - y_{\min}) \times \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right)^\gamma + y_{\min}. \tag{6}
\]

The power \(\gamma\) is a number greater than zero. If the constant \(c\) is fixed, the smaller the \(\gamma\) is, the brighter the output image will be [27], [33]. Logarithmic and power-law normalization require the specification of parameters, such as \(c\) and \(\gamma\), before performing the appropriate transformations on the input image.

For both the linear and nonlinear normalization, specifying the proper parameters usually involves a time-consuming trial-and-error process. Improper parameter specification will result in poor contrast adjustment and loss of information if the
Algorithm 1: Rank-based transformation (RBT) of an image $I$ for contrast adjustment. The data type of the image is $\mathbb{D}$. The dynamic range is $[x_{\min}, x_{\max}]$ for the input, and $[y_{\min}, y_{\max}]$ for the output. Unless specified, $y_{\min}$ and $y_{\max}$ are by default set to the smallest and largest value allowed in the data type $\mathbb{D}$.

**Input:** $I \in \mathbb{D}^{n \times n}$, $[x_{\min}, x_{\max}]$, $(y_{\min}, y_{\max})$

**Output:** $I_{\text{RBT}} \in \mathbb{D}^{n \times n}$

1. Initialize $I_{\text{RBT}}$ with zeros
2. $s \leftarrow$ unique elements in $I$ sorted in ascending order
3. $u \leftarrow \text{length}(s)$
4. For $i = \{1, 2, \ldots, u\}$ do
5. $I_{\text{RBT}}(i) = s(i) = \frac{y_{\max} - y_{\min}}{y_{\max} - y_{\min}} + y_{\min}$
6. end for

Transformed intensity values exceed the available DR of the data type.

Rank algorithms refer to the procedures that rank the items in a dataset according to a given criterion [34]. Applying rank algorithms as filters combined with a moving window is a well-known method in image processing [35], [36]. The rank-based filter combined with a moving window allows for adaptive processing of local signals. In a limited region, the histogram of objects, noise, and fringes may exhibit similar patterns, making it challenging to differentiate the information and leading to difficulties in effectively adjusting the image’s contrast. [36]. Therefore, global methods are still useful in some cases.

In this study, based on the concept of global ranking pixels of an image, we propose a simple method that does not require specific parameters from the user, and avoids information loss. Considering the previously reviewed work on local ranking of pixels, we were surprised to not finding any works proposing global ranking which we do here. This method aims to help users to quickly adjust the contrast of an underexposed image and to visualize the information in the gradients of the image.

**II. METHOD AND ALGORITHM**

The rank-based transformation (RBT) method we propose is a combination of histogram expansion and normalization. The steps of the RBT method are shown in Algorithm 1. First, each pixel in the input image is assigned a “rank” ($R$) based on its intensity value in ascending order, i.e., the pixel with the smallest intensity value gets the rank 1, and so forth. [35] Pixels with the same intensity value have the same rank. The number of ranks is equal to the number of unique values ($u$) among all pixels in the input image. Each rank is reduced by 1 as $R - 1$ to shift the first rank to zero, so that the new interval starts from zero, and then divided by $u - 1$ (the number of intervals over which all ranks will be equally distributed), so that each intensity level is redistributed with equal intervals in between. Then, the evenly spaced intensity levels are multiplied by the size of the full range over which the user demands to redistribute the intensity, thus scaling the values to fit the data type. In the output image, each pixel of the input image is replaced by these scaled ranks.

To demonstrate RBT usage and compare to other methods, we use an underexposed brightfield microscopic image of yeast inside a microfluidic channel (Fig. 2a). It is a uint16 grayscale image.

For comparison, we use the linear normalization (1), the multiplicative method (2), the logarithmic normalization (4), and the power-law normalization (6). The data type of the output image is in all cases set to uint16. For the linear and power-law normalization, the target range of $y$ is $[0, 65535]$. For the methods where the upper range cannot be fixed, $y_{\min}$ is set to 0.

We demonstrate the multiplicative method by multiplying the intensity values of the original image by 8.9 and 20. When multiplying by 8.9, the new intensity range remains less than 65535, meaning that none of the information carried by the high-value pixels is lost. When multiplying by 20, values exceeding 65535 are clipped to 65535, resulting in information loss.

For logarithmic normalization, $c$ is first set to 94545 to match the range of the uint16 data type without information loss. Then $c$ is increased to 180000, the new values exceeding 65535 are clipped to be 65535, and the information carried by the high-value pixels is lost and cannot be retrieved after exporting the transformed matrix to an image.

To demonstrate power-law normalization, $\gamma$ is set to 0.7, 0.3, and 0.1. Altering $\gamma$ in (6) alters the peak of the histogram without changing the upper and lower bound of the intensity range.

We also applied the two built-in MATLAB functions histeq [37] and adapthisteq, [38], which are the functions for implementing histogram equalization (HE) and contrast-limited adaptive histogram equalization (CLAHE), respectively. The HE of the “histeq” function works on the global contrast of the image, while the adapthisteq works on the local contrast in different parts of the image. We applied the default settings of the two MATLAB functions mentioned above to show that the lack of prior knowledge about the image leads to an inappropriate choice of parameters for adjusting the image. The histeq default setting of number of points (NPTS) and number of discrete gray levels (n) are 256 and 64, respectively. The adapthisteq contains six parameters with the default setting mentioned after it: number of tiles (NumTiles) [8, 8], contrast enhancement limit (ClipLimit) 0.01, number of histogram bins (NBins) 256, range of output data (Range) full, distribution type (Distribution) uniform, and distribution parameter (Alpha) 0.4.

All of the calculations, image adjustments, and histogram generation were performed using MATLAB R2019b (MathWorks). The lower and upper bounds of the target range are determined by the data type of the image. MATLAB automatically detects the data type, so there is no need to specify it during the processing.

Of the seven methods (RBT, Linear, Multiplicative, Logarithmic, Power-law, HE, and CLAHE), the RBT method requires no parameter to be specified, unless one wishes to change data type. The other six methods, require the specification of not only the target range, but also specific parameters, such as $c$, $\gamma$, and number of tiles to obtain appropriate output (Table I).
<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter specification</th>
<th>Information loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBT</td>
<td>Linear</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Multiplicative</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Logarithmic</td>
<td>Possible</td>
</tr>
<tr>
<td></td>
<td>Power-law</td>
<td>Possible</td>
</tr>
<tr>
<td></td>
<td>HE</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>CLAHE</td>
<td>Possible</td>
</tr>
</tbody>
</table>

Cellpose is a deep-learning based, open-source generalist tool for cellular image segmentation. Cellpose was trained on a large dataset of highly varied images of cells, and does not require model retraining or parameter adjustments to precisely segment cells from a wide range of image types. [39] We also applied the Cellpose to demonstrate the impact of applying our RBT method before cell segmentation compared to directly applying the original image for further processing. We fed the original image (Figure 3a) and the image processed with RBT (Figure 3d) into Cellpose. Cellpose worked well on both images for cell outline prediction and cell morphology prediction when the cell size parameter was set to 10 pixels.

### III. Results and Discussion

The chosen example image is a typical underexposed image of yeast cells in a microfluidic channel (Fig. 2a) with the value bins representing the intensity distribution of the pixels clustered on the left side of the histogram (Fig. 2b).

The image after RBT processing (Fig. 2c) shows a histogram (Fig. 2d) that was expanded to the full DR of the data type. The intensity bins in the original image were expanded from [919, 7323] to [0, 65535] after RBT processing. The RBT method non-linearly normalizes all intensity values in the original image. The entire DR is divided into \((u-1)\) intervals, and the intervals between each intensity level is made equal. No parameters need to be manually specified during RBT, and no information loss occurs because each intensity level in the original image is preserved and expanded over the full available DR of the data type.

Linear normalization remapped the intensity levels to the full available DR (Fig. 2e), resulting in a histogram (Fig. 2f) that looks similar to the RBT histogram (Fig. 2d). However, the intervals between the rescaled values are different because the linear normalization does not redistribute the intensity values with equal intervals.

Multiplying the original image by 8.9 (Fig. 2g) resulted in no loss of the information carried by the high-value pixels (Fig. 2h), because the largest new value does not exceed 65535. Multiplying the original image by 20 (Fig. 2i) clipped all of the pixels with new values exceeding 65535 to be 65535, resulting in the spike at the right end of the histogram (Fig. 2j). The information lost can not be retrieved after exporting intensity values as a new image. Simply multiplying the intensity values of an underexposed image could increase the contrast by increasing the intervals between the intensity values. However, this method results in information loss if the intensity values exceed the available DR of the data type.

For the logarithmic normalization (4), the standardization of the logarithms to [1, 2] makes the brightness of the output image dominated by \(c\). To match the DR of the \(\text{uint16}\) data type, we set \(c\) to 94545, so the largest intensity value does not exceed 65535 (Fig. 2j). This produces a brighter image (Fig. 2k) than the original input. We can increase the value of \(c\) to produce brighter output images like Fig. 2m. However, due to the limited DR of \(\text{uint16}\), the rescaled values that exceed 65535 are then clipped and cause the spike at the right end of the histogram (Fig. 2n).

For power-law normalization (6), \(\gamma\) alters the position of the peak center of the histogram (Fig. 2p, 2r) and thus the brightness of the output image (Fig. 2o, 2q). The smaller the \(\gamma\), the brighter the output image will be, because the peak center of the histogram will be shifted to the right. However, if \(\gamma\) is too small, the value bins will cluster on the right side of the histogram (Fig. 2t), resulting in an overexposed output image (Fig. 2u). Therefore, identifying an appropriate \(\gamma\) value that can shift the peak while maintaining the maximum half-peak width is key to properly normalizing and expanding the histogram with power-law normalization.

The image processed with the MATLAB function \(\text{histeq}\) to perform HE with default settings (Fig. 2v) resulted in a histogram with very few bins (Fig. 2w) and loss of most of the details in the original image. The information loss was due to the fact that \(\text{histeq}\) algorithm redistributed the intensity values into fewer bins than the original image. An advantage is that noise in the original image is also filtered out. If we specify a larger number of discrete gray levels, more information is preserved.

The image processed with the MATLAB function \(\text{adaphisteq}\) to perform CLAHE with default settings (Fig. 2x) resulted in a histogram that is improperly expanded (Fig. 2v). It is not a good contrast adjustment in the original image since the image remain underexposed and it is harder to see the cells than in the original image. Without prior knowledge, there is little chance of setting the proper parameters when first applying this algorithm to properly adjust the image contrast.

To evaluate whether RBT preprocessing makes a difference in cell segmentation with Cellpose, we fed both the original image (Fig. 3a) and the image after RBT preprocessing (Fig. 3d) to Cellpose, and obtained the predicted cell contours (Fig. 3b and Fig. 3c) and predicted cell morphology (Fig. 3e and Fig. 3f), respectively. Comparing the circled area in Fig. 3c and Fig. 3f, one see that without RBT preprocessing, some cells were missed and the background noise (marked by the arrow in Fig. 3c) was increased. With RBT preprocessing, the cell contours in the circled area were better captured as shown in Fig. 3e and only a small part of the microfluidic channel was captured as an interfering part marked by the arrow in Fig. 3f). These results indicate that preprocessing with RBT could benefit the image segmentation of state-of-the-art machine learning tools.

### IV. Conclusions

Rank-based transformation (RBT) is a simple histogram expansion method for image contrast adjustment without in-
formulation loss. The RBT algorithm normalizes and equalizes all the intensity gradients of the image over the entire intensity range of the image data type, giving equal weight to all gradients. The information carried by each pixel is preserved during the processing. RBT can be the first tool to apply to adjust the contrast of underexposed images, since no specific parameters need to be decided before applying this algorithm. Our tests with the AI cell segmentation tool showed that the RBT image generates better segmentation results than the original image. These results indicated that our RBT algorithm can help improve the performance of the AI segmentation tool by adjusting the contrast of the image.

ACKNOWLEDGMENTS

We gratefully acknowledge the funding from the Ministry of Science and Technology in Taiwan (MOST 108-2811-E-006-046, 109-2224-F-006-003, 110-2326-B-006-010, 111-2221-E-006-186). We thank Prof. Diego di Bernardo, Dr. Filippo Melonacina, and Dr. Gianfranco Fiore for generously sharing and explaining the image data. We thank our colleagues Yu-Heng Wu and Jose Ramon Chang for their help with the computations.

REFERENCES


...
Fig. 2. Original and processed images (left) with their histograms (right). The original uint16 grayscale image (a) is underexposed and its intensity bins are clustered on the left side of its histogram (b). After RBT processing, the image (c) shows a normalized histogram (d) with equal intervals between each intensity level. After linear normalization, image (e) shows a histogram (f) similar to (d), but with unequal intervals between intensity values. Direct multiplication of the intensity values of the original image by $255.9$ produces image (g) with an expanded histogram (h) whose maximum value does not exceed the DR of the uint16 data type. Direct multiplication by $20$ produces image (i), and on its histogram (j) the spike at the right end represents the values exceed 65535 are clipped to 65535. Logarithmic normalization with $c = 94545$ produces image (k), whose histogram (l) is similar to (d), with no loss of high-order pixel information. Increasing $c$ to 180000 produces image (m), whose histogram (n) has a spike at the right end, indicating that the values exceed 65535 are clipped to 65535. Power-law normalization with $\gamma = 0.7$ produces image (o), the peak of its histogram (p) is at 27750. When $\gamma = 0.3$ and produces image (q), the peak of its histogram (r) shifts to 41650. When $\gamma = 0.1$ and produces image (s), the peak of its histogram (t) shifts to 50340. HE performed with default settings of the MATLAB histeq function produces image (u) and its histogram (v) with fewer bins loses most of the gray level details. CLAHE performed with the default settings of the MATLAB adapthisteq function produces image (w) and its histogram (x) indicates the image is improperly transformed and results in poor contrast adjustment. The bin width of each histogram is set to 1.
Fig. 3. Comparison of using the original (a) and the RBT image (d) as input to the state-of-the-art cell segmentation tool “Cellpose,” with the average cell diameter set to 10 pixels. The predicted cell contour is shown in (b) and (e), and the predicted cell morphology is shown in (c) and (f) for the original and RBT image, respectively. The square in (b) and (e), and the circle in (c) and (f) highlight the regions with visible differences. The arrow in (c), (e) and (f) indicates the background noise caused by the microfluidic channels. The RBT image input provides better quality with more accurate cell contour predictions and less undesired background noise.