Oregon Undergraduate Research Journal
24.2 (2026) ISSN: 2160-617X (online)
ourj.uoregon.edu

Histropy: A Python-Based Computer Program for Quantifications

Sagarika Menon* and Peter Moeck**

Abstract

Histropy is an interactive Python-based computer program for the quantification of selected features of two-dimensional (2D) images/patterns by means of calculations based on the pixel intensities in this data, their histograms, and user-selected sections of those histograms. As is the standard, the histograms of these images display pixel-intensity values along the x-axis of a 2D Cartesian plot, with the frequency of each intensity value within the image represented along the y-axis. Histropy generates an image's histogram surrounded by a graphical user interface that allows one to select any range of image-pixel intensity levels, i.e., sections along the histogram's x-axis, using either the computer mouse or numerical text entries. The program calculates the so-called Monkey Model Shannon entropy and root-mean-square contrast for the selected section and subsequently displays them as part of what we call a histogram-workspace-plot in a computer window. To support the visual identification of small peaks in the histograms, the user can switch between a linear and a log base 10 scale display for the y-axis. Pixel intensity data from different images can be overlaid onto the same histogram-workspace-plot for visual comparisons. Histogram-workspace-plots can be saved in PNG format for future use. The source code of the program and a brief user manual are published on GitHub (Author1, 2024). While the appendix briefly highlights how the program supports the distinction between crystallographic symmetries and pseudosymmetries, this paper focuses on the computer program itself. The reader is encouraged to install the program and test its utility for other applications.

1. Introduction

The term "histogram" was coined around 1892 by statistician Karl Pearson to describe the visual representation of the distribution of data that quantifies the frequency of data points that fall into "mutually disjoint subsets called buckets" (Ioannidis, 2003). Histograms were thereby "conceived as a visual aid to statistical approximations." The visual analysis that a histogram facilitates, combined with the quantitative information that can be extracted from it, gives histograms a wide range of applications, ranging, for example, from analyzing the distributions of test scores in a classroom to probabilistically characterizing the behavior of river discharge.

Another typical application of histograms is the analysis of digital images, used in fields ranging from microscopy (Wood, 2026) to

**Sagarika Menon (smenon92@gatech.edu) is the creator of the computer program and is in the Nano-Crystallography Group at Portland State University. She is currently a second year student at the Georgia Institute of Technology pursuing Bachelors of Science degrees in physics and computer science. Her research interests include crystallography and 2D material fabrication. Upon graduation in May 2028, she plans to pursue a Ph.D. in experimental condensed matter physics.*

*** Peter Moeck (pmoeck@pdx.edu) leads the Nano-Crystallography Group at Portland State University. He is an applied crystallographer who has, in recent years, pioneered the application of a geometric form of information theory to crystallographic symmetry classifications of experimental data with two spatial dimensions.*

computer vision. Pixel intensity histograms, which represent the distribution of the intensities of all pixels in an image, provide a measure of the image that can be useful for identifying similar images, compressing the image, and more. This paper introduces Histropy, a simple computational tool for digital image analysis of 2D grayscale images. The computer program Histropy generates a pixel intensity histogram for images in grayscale, Red-Green-Blue color (RGB), or a uniform hue with a color-tone-range. Histropy enables the numerical analysis of user-selected bin-ranges within the histogram. A bin-range is defined as a subsection of the image histogram identified by upper and lower bounds on the pixel values. In other words, it is a selection of two values along the histogram's x-axis that defines a range of the histogram to analyze. One of the name-giving features of the program is the calculation of the so-called "Monkey Model" Shannon entropy (Razlighi & Kehtarnavas, 2009) of 2D images. Shannon entropy, or informational entropy, measures the "randomness" of outcomes of a given random variable (Shannon, 1948) and is defined as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i).$$

In that equation, $P(x_i)$ refers to the probability of random variable X taking the value of x_i. We sum over all possible outcomes of this variable, resulting in H, which is measured in bits due to the use of the logarithm to the basis of two. This definition is notably difficult to expand into two dimensions (Razlighi & Kehtarnavas, 2009). Thus, for our purposes, Histropy considers the histogram of a 2D image and calculates the one-dimensional entropy using the distribution of grayscale pixel values. For this calculation, we substitute the probability $P(x_i)$ in (1) with the quotient of the number of pixels with a certain gray level within a certain histogram bin (of unit size for simplicity) and the total number of pixels in the bin-range. The sum in (1) is then over all bins in that range. This one-dimensional entropy is the aforementioned "Monkey Model" Shannon entropy of 2D images.

The computer program's motivating prospective use is to quantitatively distinguish between symmetries and pseudosymmetries in a series of noisy and noise-filtered 2D-periodic images, i.e., crystal patterns. The noise-filtering of experimentally obtained and synthetic crystal patterns will involve crystallographic image processing (Zou et al., 2016) after objective, i.e., information-theory-based (Kanatani, 1998), crystallographic symmetry classifications (Author2, 2022). The appendix briefly showcases an example of a highly pseudosymmetric crystal pattern and some of the characteristics of that pseudosymmetry that Histropy can assist in visualizing.

For crystallographic gray-scale patterns such as the one shown in the appendix in the background of Fig. A1a, where there are a few pronounced peaks in the corresponding histogram (Fig. A1b), it will be highly informative to make quantifying calculations for selected ranges of the pixel intensity values. Competing computer programs, e.g., the histogram routines that are part of the well-known and widely used electron crystallography software package CRISP (Zou et al., 2016), do not typically offer the functionality desired for such studies, especially the functionality of overlaying image histograms for comparison. This feature, as shown in the appendix, allows for the user to directly compare periodic images with certain enforced plane group symmetries. Not only does this clearly illustrate the removal of noise from the image, but it also allows for the identification of cases in which the enforced symmetry is not actually reflected in the original image. In the future, Histropy may be used to assist in characterizing lab-grown materials, especially when the material being grown has different phases that can be recognized through the identified symmetry.

Parts of this code could be reused for other 2D image histogram applications that arise in the future. Because the source code of this program is freely available on GitHub, a web-based platform for sharing and collaborating on open-source code (Author1, 2024), other researchers are invited to download and modify it to support their own studies (that do not need to be based on information in two-dimensional crystal patterns or gray-scale images). For instance, the interactive histogram interface and overlay capabilities could be useful for analyzing spectral imaging data in astrophysics, where researchers isolate specific wavelength ranges to study particular elements. The bin-range selection feature would enable targeted analysis of specific spectral features, and overlays would allow comparison of the same feature across different image processing approaches. In general, similar applications could be found in any field where quantitative comparison of pixel intensity distributions within specific value ranges is valuable.

2. Methods

2.1. Design Requirements and Purpose

The development of Histropy was motivated by the development of an information-theoretic approach to crystallographic image analysis (Author2, 2022). The initial design aimed to satisfy four main goals.

1. Display Shannon entropy calculations as a measure of information content in crystal patterns, utilizing the one-dimensional "Monkey Model" approach applied to 2D image histograms.
2. Implement an interactive histogram interface that would enable elementary image segmentation through user-selected bin-range analysis, allowing selective quantification of specific histogram features.
3. Include a histogram overlay functionality to facilitate direct comparison between multiple image versions (e.g., before and after symmetry enforcement or noise filtering).
4. Make our software open-source in order to allow for adaptations and extensions to be built on top of it.

These capabilities were selected to address a gap in existing crystallographic software, which lacks the second and third goals we highlighted that we found very useful in the detailed analysis of pseudosymmetric patterns. Our goal was to create a tool to aid in the quantitative analysis of crystallographic image processing results by evaluating the effects of symmetry enforcement on noisy crystal patterns.

2.2. Development Process

Development initially began with the backend system to run calculations (Shannon entropy, root-mean-square (RMS) contrast, total intensity, etc.). The correctness of these calculations was verified using controlled test images with known properties. These included monochrome images and images with Gaussian noise of known variance, ensuring mathematical accuracy before proceeding to real-world applications.

With the computational methods validated, we developed the graphical user interface using Python's Matplotlib library for interactive histogram visualization. During this phase, we identified the need for logarithmic (log-base-10) y-axis scaling to better visualize minor histogram features.

Along the way, we conducted tests on a crystallographic dataset to generate the examples presented in the Appendix and to verify the program's functionality for our research group's intended purpose. Throughout development, we emphasized simplicity and intuitive use (verified by extensive user testing) to ensure accessibility to a broad audience. We aimed to create a low barrier to entry while maintaining analytical rigor.

2.3. Implementation Details

Python was chosen as the development language since it has extensive support for interactive plots and offers ease of adaptability and extension. Given our intention to release Histropy as open-source software, these characteristics aligned well with our goal of broad accessibility. The Histropy package can be run on any operating system (e.g. MacOS, Windows) and consists of two primary classes. One class handles all pixel-value calculations, including entropy, RMS contrast, and statistical measures (this was developed in our initial computational verification phase), while the other manages user interactions and workspace-plot updates. The interactive histogram functionality was implemented using Matplotlib's event handling system, which provides responsive real-time updates as users adjust parameters or select histogram regions.

Images inputted into Histropy can be arbitrarily sized and must be of 8-bit or 16-bit information depth. We recommend up to 1024 pixels maximum on both sides, as larger images slow the program's performance. Performance characteristics vary with image bit-depth. When analyzing 8-bit images, Histropy runs all actions in one to five seconds. The analysis of 16-bit images is about fifteen times slower on average, with most calculations taking around thirty seconds. These analysis times will increase as the user overlays more histograms. These timings were taken by running Histropy on a MacBook Pro with an Apple M3 processor with a 4.05 GHz max CPU clock rate as a part of our testing process.

3. Description of the Computer Program

3.1. 2D Image Uploads and the Main Histogram-Workspace-Plot

As implied by the title of the paper, Histropy's standard input/upload option is grayscale images with two spatial dimensions. Note that Histropy can only handle images with 16-bit depth when they are grayscale, so 16-bit color images must be converted before being uploaded to Histropy. Such a conversion can, for example, be done using the open-access program GIMP (The GIMP Development Team, 2019) by going to the "Image" dropdown menu and then selecting "Grayscale" for the "Mode". When the user selects an 8-bit color image to view/analyze through Histropy, the program converts the image to grayscale using the following equation to compute the intensity value for each pixel based on the pixel's RGB value:

$$Grayscale\ Value = (0.299 \cdot R) + (0.587 \cdot G) + (0.114 \cdot B),$$

where R stands for red, G for green, and B for blue. The coefficients reflect the fact that human eyes are most sensitive to green light and least sensitive to blue.

Histropy's histogram-workspace-plot consists of the histogram of pixel intensities of a user-selected 2D image (which does not need to be a crystal pattern), a toolbar at the bottom of the screen that allows the user to navigate the histogram, four selection spaces to the right of the histogram that allow the user to quantify the histogram and image's information, and a "thumbnail" display of the image being plotted to the right of the selection spaces, Fig. 1. Note that the "Histropy window" must be viewed in full-screen mode for the layout seen in that figure. The following subsections will explore the purpose and function of each of the four selection spaces of the program's user interface.

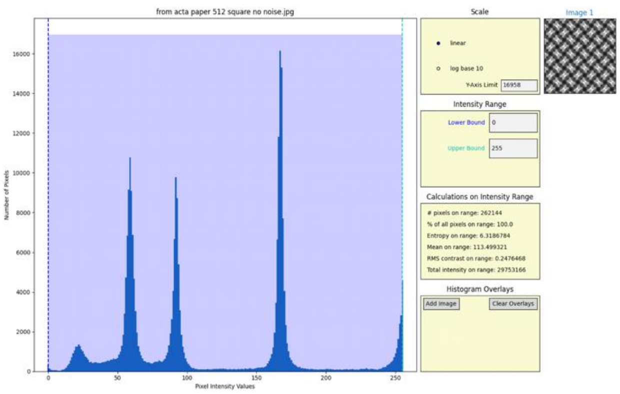


Figure 1. A screenshot of Histropy's workspace after the user has uploaded an image (in either JPG/JPEG, PNG, GIF, BMP, or baseline TIF/TIFF formats). Note that the "thumbnail" of the uploaded image is displayed for easy reference in the top right corner and that the image title, "Image 1," has a font color corresponding to the color of its histogram. In the following, we will make, for the sake of simplicity, no strict distinction between such thumbnail representations and the actual images that were uploaded. Also, note that the histogram's title contains the file name of the selected image. This particular histogram is from a 512 by 512 pixel cutout of the crystal pattern in the background of Figure A1a in the appendix. This figure is displayed in color in the online version of this paper.

3.2. Selection Space 1: Scale

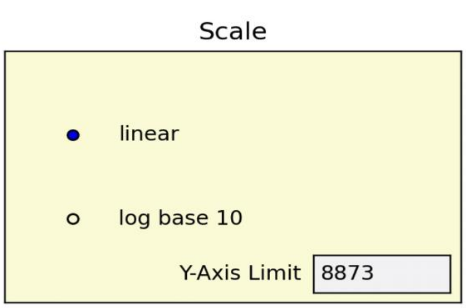


Figure 2. The scale selection space set to linear with the y-axis limit is set to the default (max value of the inputted image).

The first selection space, "Scale," is reproduced in Fig. 2 and allows the user to toggle between a linear and log-base-10 scale of the y-axis of a histogram. By including an option for log-base-10, we can make features of the histogram present for less frequently occurring pixel values while still visualizing the full distribution. A linear scale can suppress these features due to the domination of more common pixel values, so this option is vital to analyzing small features of our image histogram.

This selection space also contains an input field for a y-axis limit, which defaults to the maximum y-value in the histogram. The default value represents how many pixels have the most common pixel intensity value, i.e., the height of the most prominent peak in the histogram.

3.3. Selection Space 2: Intensity Range

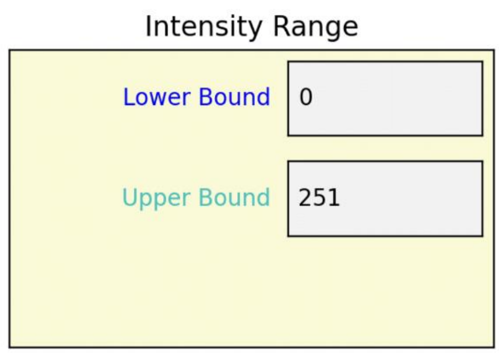


Figure 3. The selection space for the range of pixel intensities (x-values on the histogram) that the calculations are computed over. The text fields shown here are editable directly by clicking on them and typing in new values.

The numerical fields in the second selection space, "Intensity Range," displayed in Fig. 3, set the range for calculations performed in the third selection space. These fields default to the uploaded image's minimum and maximum pixel intensity. They can be set by directly typing into the text fields themselves or by clicking directly on the histogram, which will automatically set the range in the selection space fields to the x-value corresponding to where the user's mouse is pointing. When the user's mouse hovers over the histogram, the numerical entries in the bottom right corner of the screen will display the coordinates that the mouse is over, Fig. 4. These coordinates can be used to accurately select the x-values for the range. The range selected is visually represented by the vertical bars and a translucent blue rectangle shown on the histogram, Fig. 4. This range selection allows the user to perform an elementary form of segmentation by separating out the pixel buckets that contribute to a certain histogram peak. In a periodic image, this segmentation singles out a single repeating feature and allows the user to selectively identify that feature's noisiness and variance via the entropy and RMS contrast calculations described in the next subsection. Consider, for example, the dark gray squares in Fig. A1a. Additionally, when comparing multiple processed versions of the same image, one can narrow in on the impacts of that processing on a single repeating feature by comparing its entropy and RMS contrast values before and after processing.

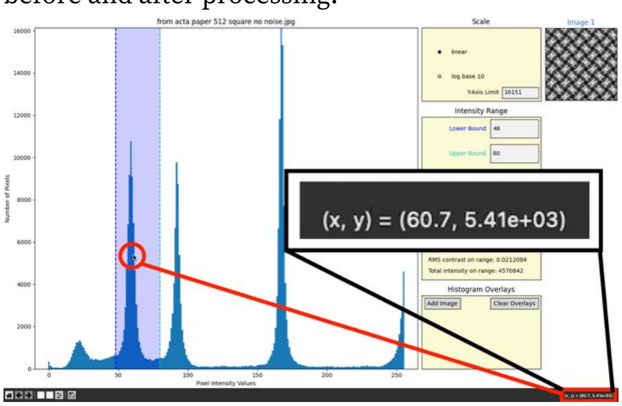


Figure 4. The cursor circled in red on the histogram here corresponds to the coordinates shown in the bottom right corner in a red box. This box is also blown up to show the coordinate values. The red box and line indicate the connection between the cursor placement and the value displayed in the bottom right corner. Upon clicking at this coordinate, the lower bound of the intensity range would update to 61 (where the cursor is pointing), and this change would reflect in the selection space text box. This figure is displayed in color in the online version of this paper.

3.4. Selection Space 3: Calculations

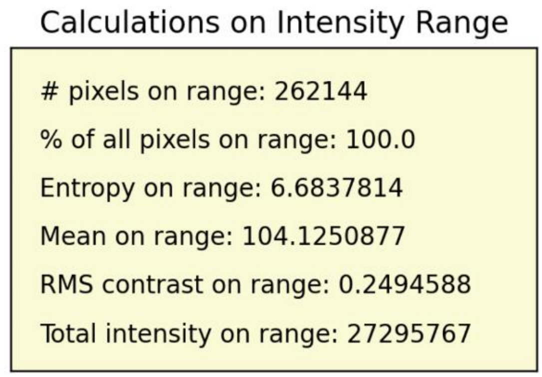


Figure 5. The calculation selection space's read-only text displays will update as the intensity range changes.

The third selection space, "Calculations," shown in Fig. 5, is a display of the results of the following calculations over a user-selected range (which defaults to the entire x-axis range of the histogram). This selection space gives:

1. the number of pixels on the given range,
2. the percent of total pixels that are found in the given range,
3. a simple form of the Shannon entropy on the given range,
4. the mean on the given range,
5. the RMS contrast on the range, and
6. the total pixel intensity on the range (a sum).

These dynamic calculations are the feature of Histropy that quantifies aspects of a 2D image's histogram, helping to identify periodic features and their noisiness. Note that these calculations are not impacted by y-axis display restrictions entered in the "Scale" selection space.

In the equations below, N is the number of pixels on the range, p_i is the intensity of the i^{th} pixel, and d is $2^{bit\ depth} - 1$ (e.g. $d = 2^8 - 1$ for an 8-bit image). For a 2D image's Monkey Model Shannon entropy H in (3), b_j is the number of pixels in the j^{th} bin (the number of pixels with intensity/gray scale value j). As defined in the introduction and after the substitution of $P(x_i) = \frac{b_i}{N}$ in (1), the one-dimensional Shannon entropy is calculated using the following equation:

$$-\sum_{j=}^{d} \frac{b_j}{N} \cdot \log_2\left(\frac{b_j}{N}\right). \quad (3)$$

The mean of a grayscale range is the sum of the pixel intensities, p_i, divided by the number of pixels in that range:

$$\bar{p} = \left(\frac{1}{N}\right) \sum_{i=1}^{N} p_i.$$

A given range's total image intensity is calculated as $N \cdot \bar{p}$ and its RMS contrast is calculated using

$$\frac{1}{d}\sqrt{\sum_{i=1}^{N} \frac{(p_i - \bar{p})^2}{N}}.$$

The factor $\frac{1}{d}$ normalizes the calculation's output to values between 0 and 1.

3.5. Selection Space 4: Histogram Overlay

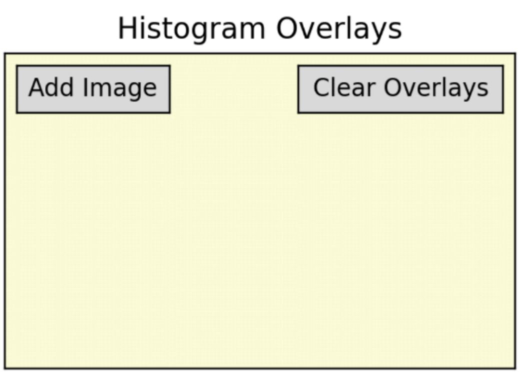
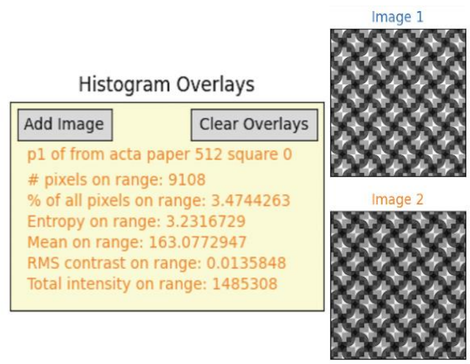


Figure 6. The Histogram Overlays selection space contains two buttons, one which opens up a file dialogue to add images to the histogram and another which resets the workspace to its initial state with no overlays.

The final selection space, "Histogram Overlays", Fig. 6, allows the user to add images whose pixel intensity data will be overlaid onto the histogram of the first image. The second image that a user uploads will have its quantifying calculation results displayed in the previously empty "Histogram Overlays" selection space, Fig. 6. This display of the calculation results will be in the color corresponding to its histogram and its thumbnail text as seen in Fig. 7 and 9.



(a) (b)

Figure 7. **(a)** The Histogram Overlays section after a second image has been loaded in and **(b)** the corresponding thumbnail section. Note the matching text colors. This figure is displayed in color in the online version of this paper.

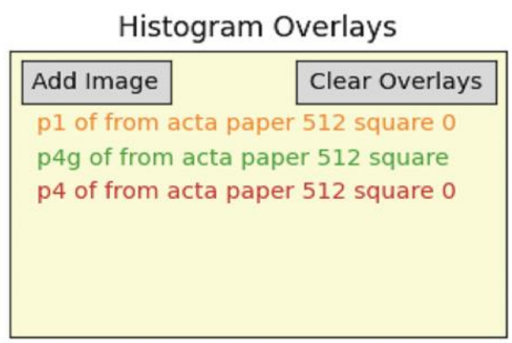


Figure 8. After 3 additional images have been read into Histropy, the selection space will display the file names of the added images in colors corresponding to how they appear on the histogram. This figure is displayed in color in the online version of this paper.

Currently, although Histropy supports overlaying several images, it is only capable of displaying the calculations present in Fig. 7a for one image in addition to the user's first image. Fig. 8 shows the display when more than one histogram is overlaid on top of the original.

In Fig. 10, three user-uploaded images (thumbnails) appear on the right underneath the first/original image, with their title colors corresponding to their appearance in the histogram-workspace-plot. Up to 22 images can be overlaid, but it is recommended to stick to 10 overlaid images or fewer (since beyond this, the program will slow down and the plot-readability is significantly reduced). If the user wishes to remove overlays, clicking on the “Clear Overlays” button in the fourth selection space, shown in Fig. 6, will reset the whole histogram-workspace-plot to what it was when the first image was loaded in, as seen in Fig. 1.

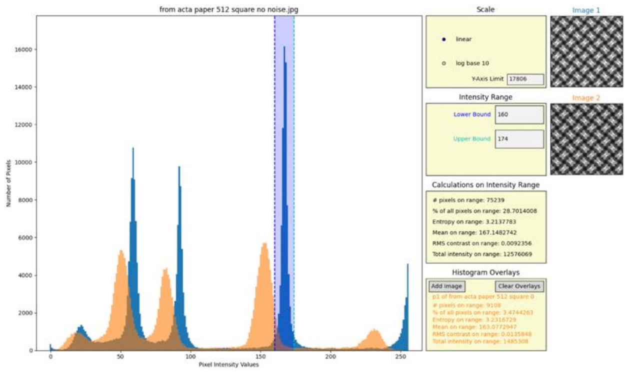


Figure 9. Two overlaid histograms, with analysis results in the “Histogram Overlays” selection space that refer to the second image. The image corresponding to the second histogram is displayed on the right under the title “Image 2.” This figure is displayed in color in the online version of this paper.

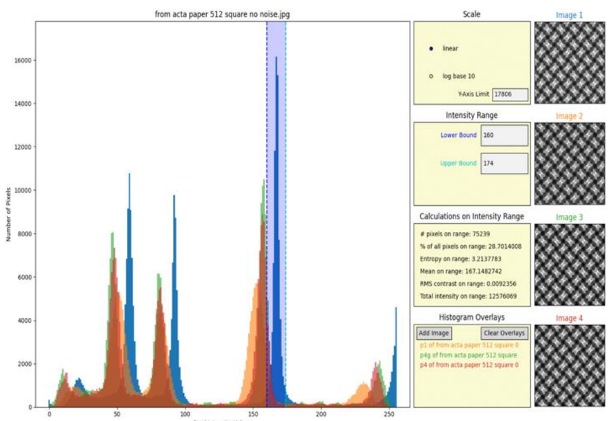


Figure 10. Four overlaid histograms with the truncated file names in the “Histogram Overlays” selection space. Note that the uploaded images on the right correspond to the overlaid histograms, with the title color of each image matching the color in which it appears in this histogram-workspace-plot, thereby matching the color of the filename that appears in the “Histogram Overlays” selection space as well. This figure is displayed in color in the online version of this paper.

3.6. Navigating the Histogram

The buttons in the bottom left corner of the histogram-workspace-plot, e.g., Fig. 1, which appear underneath the histogram, allow the user to zoom in on specific parts of the histogram using the magnifying glass and move around the viewing window along the x- and y-axes with the axes button. When the user wishes to return to a previous view, they can do so using the back arrow, or they can move forward to the next view using the forward arrow. Pressing the home button will reset all viewing changes. Finally, the user can save any populated histogram-workspace-plot as a PNG image with the save button.

4. Concluding Remarks

Histropy is a versatile computer program that facilitates the visual analysis and quantification of 2D images. The scale toggle ensures that minor histogram features remain visible alongside dominant peaks and allows for comprehensive analysis of the full pixel distribution. The bin-range selection enables users to perform elementary image segmentation by isolating specific intensity features for targeted analysis. The calculation display provides real-time quantification of Shannon entropy and RMS contrast for selected regions. Finally, the histogram overlay feature facilitates direct visual and quantitative comparison of multiple image versions. For crystallographic applications, this makes it easy to assess the effects of image processing operations and identify potential issues where enforced symmetries are not actually present in the image. The functionality of the computer program Histropy and its potential use have been briefly demonstrated in a few examples based on versions of the crystal pattern in the background of Fig. A1a in the appendix.

4.1. Limitations

While Histropy effectively serves its intended purpose of quantifying histograms, a few limitations exist that naturally motivate possible extensions discussed in the next section. The current implementation uses the one-dimensional "Monkey Model" Shannon entropy, which does not account for 2D correlated spatial information. More sophisticated entropy measures that incorporate spatial information could provide additional insights at the cost of increased computational complexity (Razlighi and Nasser, 2009).

Furthermore, while the program supports visual overlay of up to 22 histograms, calculation results are displayed for only two images simultaneously. This, along with the fact that the program slows down significantly when over 10 images are displayed, motivates modification for applications with larger image set requirements.

From an application standpoint, Histropy has been tested and validated on a limited set of crystallographic patterns, specifically variations of the pattern shown in Figure A1a.

4.2. Future Extensions

An expanded description (handbook) of this program, which details how to get started using Histropy, is freely available, together with the program's source code from Author1 (2024). This program or parts of its source code are broadly applicable to many disciplines and can be easily extended. For instance, the program's functionality could be extended by a few lines of code to support other potential uses that employ data tables with one or two dimensions in CSV format instead of only 2D images. Since Histropy's calculators rely exclusively on a discrete set of values to be plotted/run calculations over, a CSV file containing a data table could easily be loaded in and analyzed with the same tools. This could be particularly useful for applications like analyzing the data output of a scientific instrument (e.g., measuring temperature, pressure, and gas flow during material growth), comparing experimental measurements across trials, and monitoring sensor readings where quantifying and comparing the distribution of discrete values is essential.

5. Computational Details

The Histropy package is available via the Python Package Index (PyPI) and can be installed using the command **pip install Histropy**. The program can be launched with the command **python -m Histropy.Histropy**. To replicate all Histropy results illustrated in this paper, run the command **python -m Histropy.tests.Histropy_Examples**. The numerical results in this paper were obtained using Python 3.12.7 and require the installation of matplotlib, easygui, tabulate, and numpy.

Acknowledgements

This project received support from a Faculty Development Grant of Portland State University.

References

The GIMP Development Team. (2019). *GNU Image Manipulation Program (GIMP), Community, Free Software (license GPLv3)* (Version 2.10). https://gimp.org/

Ioannidis, Y. (2003). The History of Histograms (abridged). *Proceedings of the 29th International Conference on Very Large Data Bases*, 19–30. https://www.vldb.org/conf/2003/papers/S02P01.pdf

Kanatani, K. (1998). Geometric Information Criterion for Model Selection. *International Journal of Computer Vision*, *26*, 1–21.

Author1. (2024). [Details removed for blind review]. GitHub. [URL removed for blind review]

Author2. (2022). Objective crystallographic symmetry classifications of a noisy crystal pattern with strong Fedorov-type pseudosymmetries and its optimal image-quality enhancement. *Acta Crystallographica. Section A, Foundations and Advances*, *78*(3), 172–199. https://doi.org/10.1107/s2053273322000845

Author2. (2023). Genuine Plane Symmetries versus Pseudosymmetries in two Crystal Patterns of Graphic Artwork. *arXiv:2304.03915*

Razlighi, Q. R., & Kehtarnavaz, N. (2009). A Comparison Study of Image Spatial Entropy.

Proceedings of IS&T/SPIE Electronic Imaging, *7257*, pp. 72571X-1–72571X-10. https://doi.org/10.1117/12.814439

Shannon, C. E. (1948). A Mathematical theory of communication. *Bell System Technica Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Wood, C. (2026). Information Theory: An X-ray Microscopy Perspective, *arXiv: 2602.07168v1*

Zou, X., Hovmöller S., & Oleynikov, P. (2016). *Electron Crystallography*: *Electron Microscopy and Electron Diffraction*. Oxford University Press.