



OPEN ACCESS

EDITED BY

Yi-Zhe Song,
University of Surrey, United Kingdom

REVIEWED BY

Shancheng Zhao,
Jinan University, China
Ruoyi Du,
Beijing University of Posts and
Telecommunications (BUPT), China

*CORRESPONDENCE

Jon Sparring
sparring@di.ku.dk

SPECIALTY SECTION

This article was submitted to
Computer Vision,
a section of the journal
Frontiers in Computer Science

RECEIVED 09 May 2022

ACCEPTED 19 October 2022

PUBLISHED 17 November 2022

CITATION

Sparring J and Darkner S (2022) An
algebra for local histograms.
Front. Comput. Sci. 4:939563.
doi: 10.3389/fcomp.2022.939563

COPYRIGHT

© 2022 Sparring and Darkner. This is
an open-access article distributed
under the terms of the [Creative
Commons Attribution License \(CC BY\)](#).
The use, distribution or reproduction
in other forums is permitted, provided
the original author(s) and the copyright
owner(s) are credited and that the
original publication in this journal is
cited, in accordance with accepted
academic practice. No use, distribution
or reproduction is permitted which
does not comply with these terms.

An algebra for local histograms

Jon Sparring^{1,2*} and Sune Darkner¹

¹Department of Computer Science, University of Copenhagen, Copenhagen, Denmark, ²Center for Quantifying Images From MAXIV (QIM), Lyngby, Denmark

In this article, we consider local overlapping histograms of functions between discrete domains and codomains. We develop a simple algebra for local histograms. Based on a separation of overlapping domains into non-overlapping domains, we (1) show how these can be used to enumerate the size of the set of possible histograms given the local histogram domains, and (2) enumerate the number of functions, which share a specific choice of a set of local histograms. Finally, we present a decoding algorithm, which given a set of overlapping histograms, and calculate the set of functions, which share these histograms.

KEYWORDS

infinitely-additive set functions, multisets, counting histograms and functions, locally orderless histograms, reconstruction from histograms

1. Introduction

Inspired by Koenderink and Doorn (1999), we have for many years worked with images, and features derived from local histograms, and a nagging question has been, what the degrees of freedoms remain, given a set of overlapping histograms. This paper presents a theoretical investigation into the relationship between sets of local histograms and functions between discrete domains and codomains of any dimension. We describe an algebra of histograms, which is strongly related to the algebra of sets on the function domain and multisets: Given a set of local histogram's domains, $h(X_i)$, $X_i \subset X$, where X is the full domain, and X_i are subsets thereof, such that $\bigcup_i X_i = X$, we factor X into a new set of disjoint subsets $\{X'_j\}$, $\bigcup_j X'_j = X$, and with this, we are able to count the number of independent histograms, which jointly describe the total set of local histograms, and which leads to a simple countable, generative model for functions drawn from these histograms. Finally, we present a simple algorithm for generating the set of functions, which share a particular set of local histograms overlapping or not.

Our work is an extension of Sparring and Darkner (2022), where 1-dimensional signals are considered and the concept of metameric classes is introduced in the concept of local histograms. The article restricts itself to binary signals from their densely overlapping histograms. In Wu et al. (2000), the authors consider normalized histogram of images filtered with Gabor kernels (Gabor, 1946), and in particular, the limiting case of the discrete domain converging to \mathbb{Z}^2 .

This paper is organized as follows. In Section 2, we present the histogram-algebra, in Section 3 we show how the number of unique functions sharing a specific set of histograms is generated. In Section 4, we present the algorithm for calculating the set of functions, which share a given set of local histograms, and finally, Section 6 gives concluding remarks.

2. Histograms as Infinitely-Additive set functions

In the following, we will define an algebra for discrete histograms of disjoint domains, and we will extend this to non-disjoint domains by repartitioning domains.

Consider discrete domain X , co-domain A , and a functions $f : X \rightarrow A$ between them, such that the histogram $h : A \rightarrow \mathbb{Z}_+$

$$h_X(a) = \sum_{x \in X} \delta(f(x) - a), \tag{1a}$$

$$\delta(x) = \begin{cases} 1, & \text{when } x = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{1b}$$

is defined. Conceptually, we think of X as d -dimensional spatial domain $X = \{1, 2, 3, \dots, n\}^d$ with side-lengths $n > 0$, and A as an alphabet of $m > 0$ different gray values $A = \{1, \dots, m\}$, but for the properties of possibly overlapping histograms, the interpretation of the values of X and A is not important, and X and A could as well be the set {cow, cat, fish} or color triplets $\{(0, 0, 0), (0, 0, 1), \dots\}$. As long as we can define a one-to-one mapping to an index set, we need only to concern ourselves with this index.

Two key properties of a histogram are that

Property 2.1. Histograms are non-negative, $\forall a \in A h(a) \geq 0$.

Property 2.2. Every value $f(x), x \in X$ is counted once and only once.

A direct consequence of Property 2.2 is that

$$\sum_{a \in A} h_X(a) = |X|. \tag{2}$$

In this article, we are interested in counting possible histograms and for given histograms, counting the number of possible function. Let's start by examining the number of unique histograms that exists for a single domain and co-domain. Let $\mathcal{H}_X = \{h_X^i\}, \forall i, j h_X^i \neq h_X^j$ be the set of unique histograms. Its size may be calculated as unordered sampling with replacement, where we visually represent each element in X with a “•” and each bin edge with a “;.” Then the string “•••••; ; . . .” corresponds to the histogram $[1; 2; 3; \dots] \rightarrow [3; 1; 0; \dots]$. For brevity, it is convenient to assume that an ordering of the alphabet exists such that we may write the before mentioned histogram simply as $[3; 1; 0; \dots]$. The string will be $|X| + |A| - 1$ long, and all possible histograms can be produced by selecting $|A| - 1$ positions in this string for the “;” character. Thus, the number of unique histograms for a given domain X is given by the binomial coefficient,

$$|\mathcal{H}_X| = \binom{|X| + |A| - 1}{|A| - 1} = \binom{|X| + |A| - 1}{|X|}. \tag{3}$$

In the following, we will consider possibly overlapping, local histograms over the domain X . Our expositions will be divided into first non-overlapping or disjoint domains, and then we will show how overlapping domains can be repartitioned into disjoint domains, and how these relate to the original overlapping domains.

2.1. Histograms over disjoint domains

Consider a partitioning of X into $k < \infty$ disjoint subdomains $X = \bigcup_{i=1}^k X_i$, where $\forall i \neq j X_i \cap X_j = \emptyset$. Due to Property 2.2, h is a finitely-additive set function (Stover, 2022), and hence,

$$\sum_{i=1}^k h_{X_i}(a) = h_{\bigcup_i X_i}(a) = h_X(a). \tag{4}$$

As a consequence, $h_{\emptyset}(a) = 0$, and addition of histograms of disjoint domains is commutative and associative. The subtraction $h_Y(a) - h_X(a)$ is a histogram when $X \subseteq Y$, e.g.,

$$h_{X \cup Y} = h_X + h_Y \Leftrightarrow h_{X \cup Y} - h_X = h_Y \Leftrightarrow h_{X \cup Y} - h_Y = h_X, \tag{5}$$

omitting the argument a for brevity. However, subtracting any two histograms in general will likely produce negative values violating Property 2.1, and although useful at times, the result will not be a histogram.

Since the sets X_i are disjoint, the size of the set of all possible histograms of X is found by extending Equation (3) directly,

$$|\mathcal{H}| = \prod_{i=1}^k |\mathcal{H}_{X_i}|. \tag{6}$$

2.2. Partitioning of non-disjoint sets

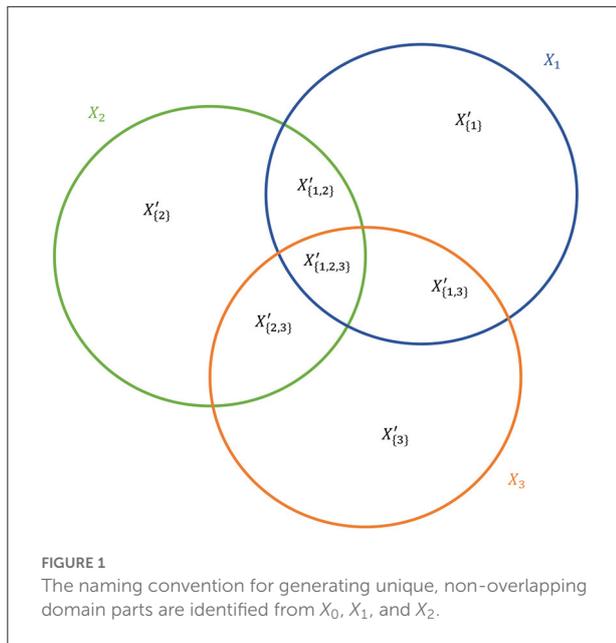
For a set of k non-disjoint domains $\mathcal{X} = \{X_i\}$ of $X = \bigcup_{i=1}^k X_i$, we can repartition X into disjoint domains of unique overlap of X_i

$$X'_I = \left(\bigcap_{j \in I} X_j \right) \setminus \left(\bigcup_{j \in \{0, 1, \dots, n-1\} \setminus I} X_j \right), I \in P_k, \tag{7}$$

where P_k is the powerset of $\{1, 2, \dots, k\}$, e.g., $P_3 = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. With this notation, we find the original sets as,

$$X_i = \bigcup_{p \in P_k : i \in p} X'_p. \tag{8}$$

Example 2.1. As an example, consider 3 sets X_1, X_2 , and X_3 , there are 7 unique intersections as illustrated in Figure 1 together with the powerset naming convention.



That is, $X'_{1,2} = (X_1 \cap X_2) \setminus X_3$ and $X_1 = X'_{1} \cup X'_{1,2} \cup X'_{1,2,3} \cup X'_{1,3}$.

Example 2.2. As a concrete example, consider the domain $X = \{1, 2, \dots, 6\}$ and the codomain $A = \{1, 2, 3\}$, and define $X_1 = \{1, 2, 3, 4\}$, $X_2 = \{3, 4, 5, 6\}$. Assuming the usual ordering of integers, we can illustrate this overlap on a line as,

$$X = \underbrace{[1; 2; 3; 4]}_{X_1} \underbrace{[3; 4; 5; 6]}_{X_2}. \tag{9}$$

Using Equation (7) we find that $P_2 = \{\{1\}, \{2\}, \{1, 2\}\}$ and that $X'_{1} = \{1, 2\}$, $X'_{1,2} = \{3, 4\}$, and $X'_{2} = \{5, 6\}$. Each of these subdomains are of size 2, and thus, the by Equation (3), number of possible histograms of each is $\binom{2+3-1}{3-1} = 6$, and the set of possible histograms is

$$\mathcal{H}_{X_i} = \{[2; 0; 0], [1; 1; 0], [1; 0; 1], [0; 2; 0], [0; 1; 1], [0; 0; 2]\}, \quad I \in P_2. \tag{10}$$

Since there are 3 disjoint regions each with 6 possible histograms, there are $6^3 = 216$ combinations of these. Introducing a natural extension of our notations on the domains to their corresponding histograms, one of these is,

$$h_{\{1\}} = h_{X'_{1}} = [1; 1; 0], \tag{11a}$$

$$h_{\{1,2\}} = h_{X'_{1,2}} = [1; 0; 1], \tag{11b}$$

$$h_{\{2\}} = h_{X'_{2}} = [0; 2; 0], \tag{11c}$$

in which case,

$$h_1 = h_{X_1} = h_{\{1\}} + h_{\{1,2\}} = [1; 1; 0] + [1; 0; 1] = [2; 1; 1] \tag{12a}$$

$$h_2 = h_{X_2} = h_{\{1,2\}} + h_{\{2\}} = [1; 0; 1] + [0; 2; 0] = [1; 2; 1]. \tag{12b}$$

Since these overlapping histograms have been generated by histograms on their disjoint parts, we are sure that a function exists on X which has histograms h_1 and h_2 . Further, since histograms are finitely-additive functions we are sure that Properties 2.1 and 2.2 are fulfilled for h_1 and h_2 .

In the following, we will count the number of functions on disjoint domains and see how these can be combined to generate the family of functions, which share overlapping histograms generated from the disjoint domains.

3. Unique functions and their histograms on disjoint domains

For a single domain X , the total number of possible functions is given as $|A|^{|X|}$, and some of these have the same histogram. Conversely, given a histogram h , the set of functions, which share this histogram can be produced as the set of distinct permutations of the function,

$$S = \underbrace{[1; \dots; 1]}_{h(1)} \underbrace{[2; \dots; 2]}_{h(2)} \underbrace{[3; \dots; 3; \dots]}_{h(3)}. \tag{13}$$

The number of distinct functions is given by

$$C_X = \prod_{i=1}^{|A|} \binom{|X| - c_X(i)}{h_X(i)}, \tag{14a}$$

$$c_X(i) = \begin{cases} 0, & i \leq 1 \\ \sum_{j=1}^i h_X(j), & \text{otherwise.} \end{cases} \tag{14b}$$

C_X is a multinomial coefficient and can be simplified to

$$C_X = \binom{|X| - c(1)}{h(1)} \binom{|X| - c(2)}{h(2)} \binom{|X| - c(3)}{h(3)} \dots, \tag{15a}$$

$$= \binom{|X|}{h(1)} \binom{|X| - h(1)}{h(2)} \binom{|X| - h(1) - h(2)}{h(3)} \dots, \tag{15b}$$

$$= \frac{|X|!}{h(1)! (|X| - h(1))!} \frac{(|X| - h(1))!}{h(2)! (|X| - h(1) - h(2))!} \frac{(|X| - h(1) - h(2))!}{h(3)! (|X| - h(1) - h(2) - h(3))!} \dots, \tag{15c}$$

$$= \frac{|X|!}{h(1)!h(2)!h(3)! \dots}, \tag{15d}$$

$$= \frac{|X|!}{\prod_{i=1}^{|A|} h(i)!}, \tag{15e}$$

where we for simplicity have neglected to write the subscript X and in the last term used that $(|X| - h(1) - h(2) - \dots - h(|A|))! = 1$. Like the simplified notation for h , we will also write C_i for C_{X_i} .

For the disjoint sets $\forall_{i \neq j} X_i \cap X_j = \emptyset$, the functions on X_i are independent on those on $X_j, j \neq i$, and may be chosen independently. Thus, number of functions sharing H is

$$C_X^{\text{disjoint}} = \prod_i C_{X_i}, \tag{16}$$

where C_{X_i} is Equation (14) applied to h_i .

Example 3.1. As an example, consider the (ordered) alphabet $A = \{1, 2, 3\}$ and the histogram $h_X = [1; 1; 2]$. Then by Equation (2) we know that $|X| = 4$. Finally using Equation (15) we find that

$$C_X = \frac{4!}{1!1!2!} = 12. \tag{17}$$

Assuming that X is a line, we can list all possible functions which has histogram h_X as,

$[0; 1; 2; 2], [0; 2; 1; 2], [0; 2; 2; 1], [1; 0; 2; 2], [2; 0; 1; 2], [2; 0; 2; 1], [1; 2; 0; 2], [2; 1; 0; 2], [2; 2; 0; 1], [1; 2; 2; 0], [2; 1; 2; 0], [2; 2; 1; 0]$.

Example 3.2. Another example, for the same alphabet as in Example 3.1 but with $h_X = [2; 0; 2]$ we follow the same procedure as in Example 3.1 to calculate $C = \frac{4!}{0!2!2!} = 6$, and the list possible functions on a linear domain X as,

$[0; 0; 2; 2], [0; 2; 0; 2], [0; 2; 2; 0], [2; 0; 0; 2], [2; 0; 2; 0], [2; 2; 0; 0]$.

Example 3.3. Continuing Example 2.2 with $A = \{1, 2, 3\}$, $X = \{1, 2, \dots, 6\}$, $X_1 = \{1, 2, 3, 4\}$, $X_2 = \{3, 4, 5, 6\}$, and $h_{\{1\}} = [1; 1; 0], h_{\{1,2\}} = [1; 0; 1], h_{\{2\}} = [0; 2; 0]$, the number of functions is computed from its non-overlapping parts are

$$C_{\{1\}} = \frac{2!}{1!1!0!} = 2, \quad C_{\{1,2\}} = \frac{2!}{1!0!1!} = 2, \\ C_{\{2\}} = \frac{2!}{0!2!0!} = 1, \tag{18}$$

Thus, the total number of functions for these specific histograms h_0 and h_1 is $C_{\{1\}}C_{\{1,2\}}C_{\{2\}} = 4$, and the functions are any combination of

$$f(X'_{\{1\}}) \in \{[1; 2], [2; 1]\}, \quad f(X'_{\{1,2\}}) \in \{[1; 3], [3; 1]\}, \\ f(X'_{\{2\}}) = [2; 2]. \tag{19}$$

One of the 4 functions, which have histograms h_1 and h_2 specified in Equation (12) is thus $f(X) = f(X'_{\{1\}} \cup X'_{\{1,2\}} \cup X'_{\{2\}}) = [1; 2; 3; 1; 2; 2]$.

Example 3.4. As a final example, consider a one-dimensional function over the alphabet $A = \{1, 2, 3\}$ and where $X = X_1 \cup X_2 \cup X_3$, $X_1 = \{1, 2, 3, 4\}$, $X_2 = \{2, 3, 4, 5\}$, $X_3 = \{3, 4, 5, 6\}$. The unique partitions are then given as,

$$X'_{\{1\}} = \{1\}, \quad X'_{\{1,2\}} = \{2\}, \quad X'_{\{1,3\}} = \emptyset, \quad X'_{\{1,2,3\}} = \{3, 4\}, \\ X'_{\{2\}} = \emptyset, \quad X'_{\{2,3\}} = \{5\}, \quad X'_{\{3\}} = \{6\}. \tag{20}$$

The possible histograms of the singleton domains are

$$h_I \in \{[1; 0; 0], [0; 1; 0], [0; 0; 1]\}, \quad I \in \{\{1\}, \{1, 2\}, \{2, 3\}, \{3\}\}, \tag{21}$$

and for $X'_{\{1,2,3\}}$,

$$h_{\{1,2,3\}} \in \{[2; 0; 0], [1; 1; 0], [1; 0; 1], [0; 2; 0], [0; 1; 1], [0; 0; 2]\}, \tag{22}$$

since $|X'_{\{1,2,3\}}| = 2$. The total number of different histograms is,

$$|\mathcal{H}| = \binom{3}{2}^4 \binom{4}{2} = 486. \tag{23}$$

To generate a set of functions and overlapping histograms, we choose a specific set of h_I ,

$$h_{\{1\}} = [0; 0; 1], \quad h_{\{1,2\}} = [0; 1; 0], \quad h_{\{1,2,3\}} = [1; 1; 0], \\ h_{\{2,3\}} = [0; 0; 1], \quad h_{\{3\}} = [1; 0; 0], \tag{24}$$

and thus, $h_1 = h_{\{1\}} + h_{\{1,2\}} + h_{\{1,2,3\}} = [1; 2; 1]$, $h_2 = h_{\{1,2\}} + h_{\{1,2,3\}} + h_{\{2,3\}} = [1; 2; 1]$, and $h_3 = h_{\{1,2,3\}} + h_{\{2,3\}} + h_{\{3\}} = [2; 1; 1]$. The number of functions is computed from its non-overlapping parts,

$$C_{\{1\}} = 1, \quad C_{\{1,2\}} = 1, \quad C_{\{1,2,3\}} = 2, \quad C_{\{2,3\}} = 1, \quad C_{\{3\}} = 1, \tag{25}$$

Thus, the total number of functions for these specific histograms $\mathcal{H} = \{h_1, h_2, h_3\}$ is $C_{\{1\}}C_{\{1,2\}}C_{\{1,2,3\}}C_{\{2,3\}}C_{\{3\}} = 2$, and the functions are any combination of

$$f(X'_{\{0\}}) = [2], \quad f(X'_{\{0,1\}}) = [1], \quad f(X'_{\{0,1,2\}}) \in \{[0; 1], [1; 0]\}, \\ f(X'_{\{1,2\}}) = [2], \quad f(X'_{\{2\}}) = [0], \tag{26}$$

and one of the two possible functions sharing \mathcal{H} is thus $f(X) = [2; 1; 0; 1; 2; 0]$.

In the above, we have given a method for generating histograms and functions by repartitioning the domain into disjoint domains. In the following, we will investigate how to find the set of functions, which share a set of overlapping histograms.

4. Unique functions from overlapping histograms

For a set of overlapping histograms, $\mathcal{H} = \{h_1, h_2, \dots, h_k\}$ we have yet to find a closed form solution for counting the number of functions, which share \mathcal{H} . However, by repartition their domain using Equation (7) giving $|P_k|$ disjoint domains, we are able to recursively calculate the sets of histograms for the repartitioned domains which agree with \mathcal{H} . For each domain, we have $\kappa_{h_I}(1, |X_I|), I \in P_k$ different histograms where κ is given recursively as,

$$\kappa_h(j, k) = \begin{cases} 1, & \text{if } k = 0, \\ \sum_{i=l(j,k)}^{u(j,k)} \kappa_h(j+1, k-i), & \text{otherwise,} \end{cases} \tag{27a}$$

$$u(j, k) = \min(h(j), k), \tag{27b}$$

$$l(j, k) = \max\left(0, k - \sum_{i=j+1}^{|A|} h(i)x\right). \tag{27c}$$

Example 4.1. For example, given two overlapping subdomains X_1 and X_2 , we repartitioning the domain using Equation (7) into $X'_{\{1\}} = X_1 \setminus X_2, X'_{\{1,2\}} = X_1 \cap X_2,$ and $X'_{\{2\}} = X_2 \setminus X_1.$

Further, if $A = \{1, 2\}, X'_{\{1\}} = \{1, 2\},$ and $X'_{\{1,2\}} = \{3, 4\},$ then there are the following possible combinations of histograms for h_1 and $h_{\{1\}}$:

$$h_1 = [4; 0] \Rightarrow h_{\{1\}} = [2; 0], \tag{28a}$$

$$h_1 = [3; 1] \Rightarrow h_{\{1\}} \in \{[2; 0], [1; 1]\}, \tag{28b}$$

$$h_1 = [2; 2] \Rightarrow h_{\{1\}} \in \{[2; 0], [1; 1], [0; 2]\}, \tag{28c}$$

$$h_1 = [1; 3] \Rightarrow h_{\{1\}} \in \{[1; 1], [0; 2]\}, \tag{28d}$$

$$h_1 = [0; 4] \Rightarrow h_{\{1\}} = [0; 2], \tag{28e}$$

The recursive evaluation of κ in Equation (27) for this example is visualized as the trees in Figure 2. Not that given $h_{\{1\}}$, then $h_{\{1,2\}}$ is determined directly by Equation (5) as $h_1 - h_{\{1\}}$. For example, if $h_1 = [3; 1] \wedge h_{\{1\}} = [1; 1]$ then $h_{\{1,2\}} = [3; 1] - [1; 1] = [2; 0].$

Given \mathcal{H} , we can use Equation (27) to sequentially generate a tree of histograms h_I which agree with \mathcal{H} . For example, starting with h_1 we can calculate the set of possible histograms for $(h_{\{1\}}, h_1 \setminus h_{\{1\}})$ pairs. Then for each $h_1 \setminus h_{\{1\}}$ we calculate the set

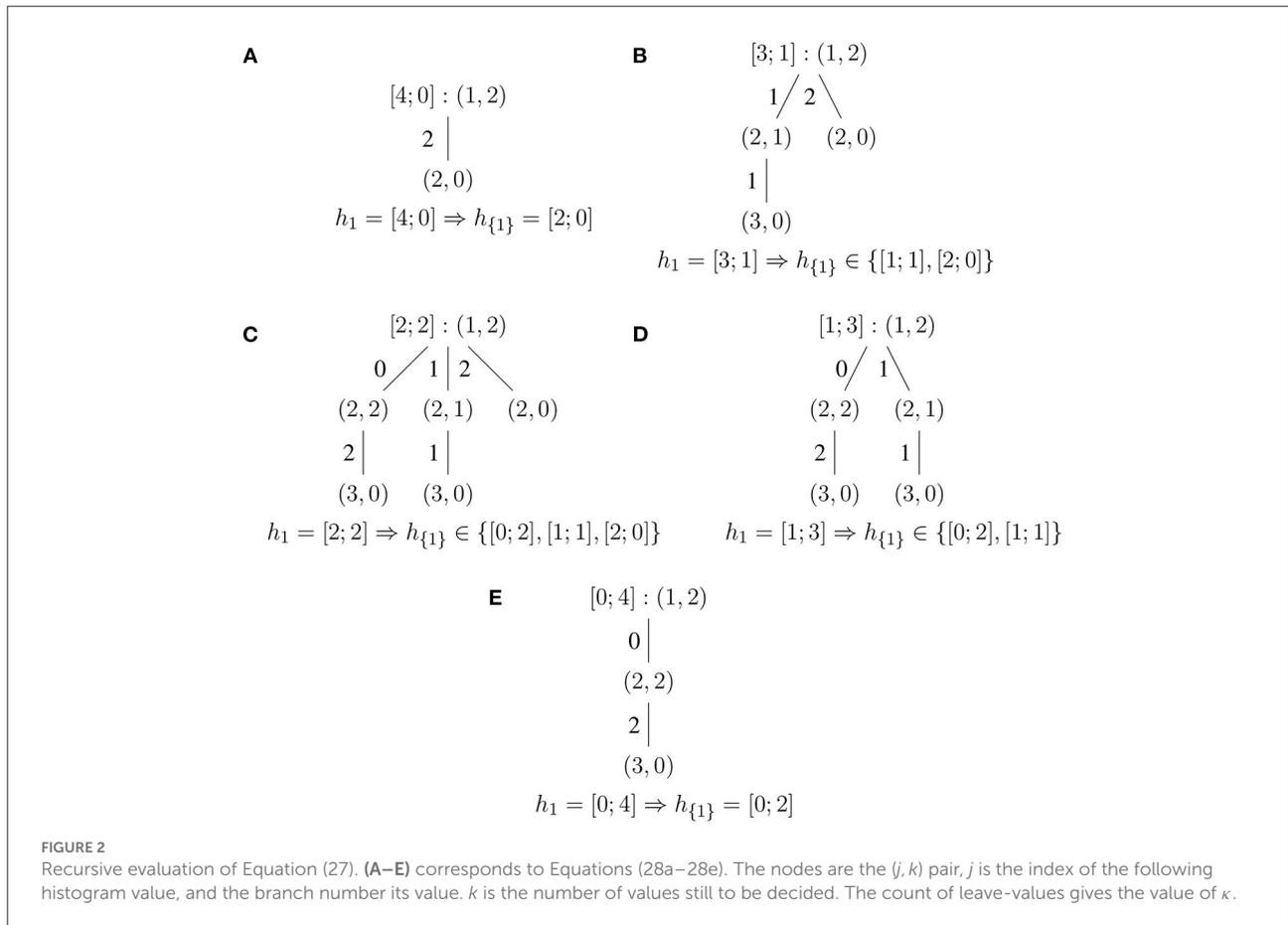


FIGURE 2 Recursive evaluation of Equation (27). (A–E) corresponds to Equations (28a–28e). The nodes are the (j, k) pair, j is the index of the following histogram value, and the branch number its value. k is the number of values still to be decided. The count of leaf-values gives the value of κ .

of possible histograms for $(h_{\{1,2\}}, h_{1 \setminus \{1\}} \setminus h_{\{1,2\}})$ pairs and so on. In practice, we have chosen to implement a sifting algorithm instead, which will be described in the following.

Given a set overlapping domains $\{X_0, X_1, \dots\}$ and their corresponding histograms $\{h_{X_0}, h_{X_1}, \dots\}$, we propose a sifting algorithm that considers a list of candidate functions that are iteratively updated as we consider additional local histograms. We produce candidate functions, and for a particular candidate function f , which has candidate values at positions $X^n = \bigcup_{i=0}^{n-1} X_i$, the next window X_n and its target histogram h_{X_n} , we identify yet to be considered region $X_n \setminus X^n$ and calculate the function

$$g_{X_n \setminus X^n} = h_{X_n} - h_{X_n \cap X^n}. \tag{29}$$

If $g \geq 0$ and $\sum_a g(a) = |X_n \setminus X^n|$, then we cannot refute the candidate, and g is the histogram of $X_n \setminus X^n$ which agrees with the histograms h_0, h_1, \dots, h_n , and hence the candidate f is replaced with a new set of candidate functions extending $f(X^n)$ with function values that have histogram g at $X_n \setminus X^n$.

The computational complexity of the algorithm is governed by the sizes of the function, the sizes of X_i , and the sweeping order of the update of the candidates. In Figure 3 are two unavoidable cases shown for a 2-dimensional domain. In the figure, X^n are denoted by the blue areas, and X_n by the green square.

Since each candidate appears to grown binomially by the size that $X_n \setminus X^n$, our experiments indicate that a sweeping order, where the cases where $X_n \setminus X^n$ is small seems to produce fewer maximum number of candidates during the reconstruction. An upper bound on the search tree is given in Section 5. The main part of our algorithm is shown in Figure 4. The full algorithm can be downloaded from github.

Example 4.2. An example of a reconstruction is shown in Figure 5, where $A = \{0, 1, 2\}$, $X = \{0, 1, \dots, 9\}^2$, and X_i are 3×3 square windows translated in both directions with a stride of 1. In this case, there are two images, which has the same set of local histograms for $m = 3$.

5. Bound on the size of the search tree

As a measure of the Computational complexity of our sifting algorithm, we will here give an upper bound on the search tree.

Given an $n \times n$ image with intensities from an alphabet A and its local histograms h_{ij} over $m \times m$ domains, X_{ij} , where $m \leq n$, and where ij is the lower left corner of the domain. We consider the maximum case of all local $(n - m + 1)^2$ histograms produced by $m \times m$ windows translated by 1 over the image domain. Our algorithm considers the histograms in a diagonal order,

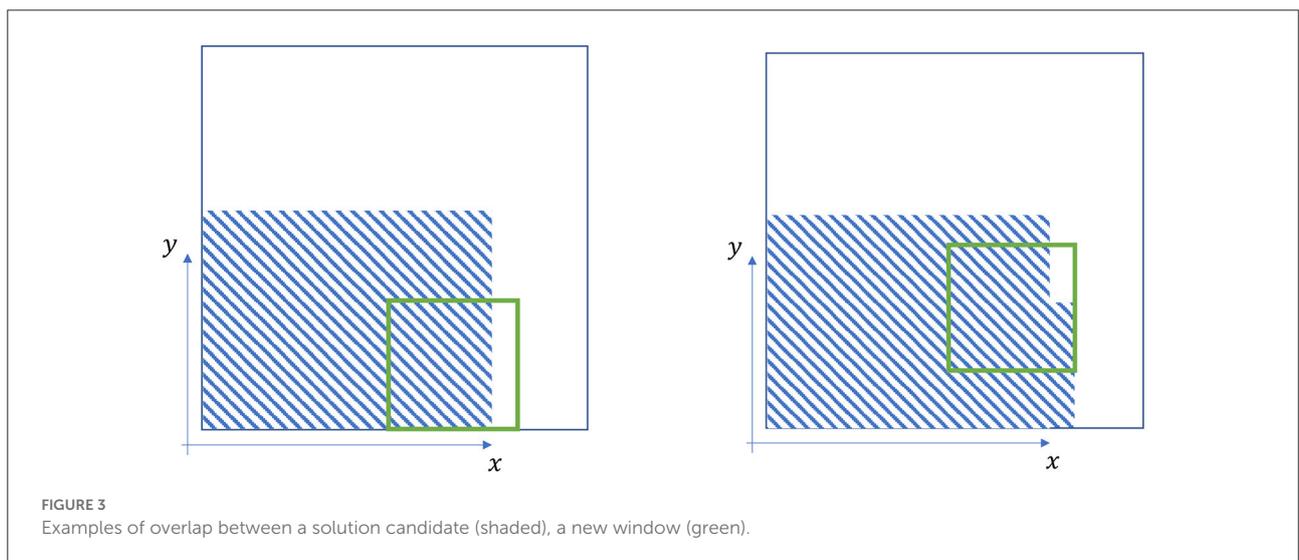
$$[h_{11}, h_{21}, h_{12}, h_{31}, h_{22}, h_{13}, h_{41}, \dots, h_{(n-m+1)(n-m+1)}].$$

Case h_{11} : Our sifting algorithm will first produce the set of candidates for X_{11} which by (15) produces $\frac{m^2!}{\prod_{i=1}^{|A|} h_{11}(i)!}$ candidates. The pseudo-uniform histogram, $|h(j) - h(k)| \leq 1, j \neq k$ maximizes this value. To prove this, consider two values in the histogram, where $h(j) > h(k)$, and the denominator,

$$d = \prod_i h(i)! \tag{30}$$

$$= h(j)!h(k)! \prod_{i \notin \{j,k\}} h(i)! \tag{31}$$

For a similiar histogram h' which is equal to h , except $h'(j) = h(j) - 1$ and $h'(k) = h(k) + 1$, then the ratio of their



```

// H - ('a option * int) [,]
//     a 2d array of all local histograms of mxm patches of
//     an nxn image. The histograms are pairs of values of
//     any type and counts

// Order matters for efficiency, here anti-diagonal ordering
let order =
  H
  |> Array2D.toList // flatten the 2d array to a list
  |> List.sortBy (fun (a,b) -> a+b)
// For all local histograms by order
List.iter (fun (i,j) ->
  let h = H.[i,j]
  // For all the candidate images in lst, iteratively collect updated
  // candidates
  lst <- List.collect (fun (org: 'a option [,]) ->
    let expansion =
      h
      // Make a list of lists of possible updates as
      // coordinate-value tuples
      |> suggest org.[i..i+m0-1, j..j+m1-1]
      // Created a list of images with update
      |> expandSuggestions org i j
    expansion
  ) lst
) order
    
```

FIGURE 4
An F# sifting algorithm.

corresponding denominator is,

$$\frac{d}{d'} = \frac{\prod_i h(i)!}{\prod_i h'(i)!} \tag{32}$$

$$= \frac{h(j)!h(k)!}{(h(j) - 1)!(h(k) + 1)!} \tag{33}$$

$$= \frac{h(j)}{h(k) + 1} \geq 1. \tag{34}$$

When $h(j) = h(k) + 1$ then $\frac{d}{d'} = 1$ otherwise $\frac{d}{d'} > 1$, and we conclude that d is minimized for pseudo-uniform histograms. Since m^2 is a constant, $\frac{m^2!}{\prod_{i=1}^{|A|} h(i)!}$ is maximized for pseudo-uniform histograms. Writing m^2 by its integer quotient and remainder,

$$m^2 = q|A| + r \tag{35}$$

where q and r are whole numbers and $0 \leq r < |A|$, then the pseudo-uniform histogram will have $|A| - r$ bins with q values and r bins with $q + 1$, and the largest number of candidates for the left-most part of the image is

$$\frac{m^2!}{\prod_{i=1}^{|A|} h_{11}(i)!} = \frac{m^2!}{q!^{|A|-r}(q+1)!^r}. \tag{36}$$

Case h_{21} : Our algorithm next considers the histogram h_{21} for the window X_{21} , which is a translated 1 wrt. X_{11} , i.e., $|X_{11} \setminus X_{21}| = |X_{21} \setminus X_{11}| = m$. When $h_{X_{11} \setminus X_{21}} = h_{X_{21} \setminus X_{11}}$, then none of the candidates generated by h_{11} can be discarded, and for each, we must consider all the additional candidates for $h_{X_{21} \setminus X_{11}}$.

In the worst case, $h_{X_{21} \setminus X_{11}}$ is pseudo-uniform. Writing m in terms of its integer quotient and remainder,

$$m = p|A| + s \tag{37}$$

where p and s are whole numbers $0 \leq s < |A|$, this gives us

$$\frac{m!}{\prod_{i=1}^{|A|} h_{X_{21} \setminus X_{11}}(i)!} = \frac{m!}{p!^{|A|-s}(p+1)!^s}. \tag{38}$$

additional hypotheses to consider for each existing candidate.

Case h_{12} : Having reach this histogram, all candidates agree with h_{11} and h_{21} . Since, $|X_{12} \setminus (X_{11} \cup X_{21})| = |X_{12} \setminus X_{11}| = m$, the number of additional hypotheses for each candidate are the same as derived for case h_{21} .

Case h_{22} : Having reach this histogram, all candidates agree with $h_{11}, h_{21}, h_{12}, h_{31}$. Since, $|X_{22} \setminus (X_{11} \cup X_{21} \cup X_{12} \cup X_{31})| = |X_{22} \setminus (X_{11} \cup X_{21} \cup X_{12})| = 1$, and there is at most one solution for this solution corresponding to a non-negative value in difference between $h_{22} - h_{X_{22} \cap (X_{11} \cup X_{21} \cup X_{12})}$. If this histogram difference is has negative value, then the candidate solution can be discarded, however, for simplicity's sake, we will ignore this. Thus, this case does not give additional candidates.

Bound on the number of candidate images: By the anti-diagonal order, we 1 time are in Case h_{00} , $n - m$ times in Case $h_{i,1}, i > 1$ and in Case $h_{1,j}, j > 1$, which are identical to Cases $h_{2,1}$ and $h_{1,2}$, and $(n - m - 1)^2$ times in Case $h_{i,j}, i, j > 1$, which are identical to Case h_{22} . Thus, we conclude that the

```

Original image:
[[0; 0; 0; 2; 0; 0; 0; 2; 1; 2]
 [0; 2; 0; 0; 0; 0; 1; 0; 1; 0]
 [0; 2; 1; 0; 2; 2; 0; 1; 0; 2]
 [0; 0; 2; 2; 0; 0; 2; 0; 2; 0]
 [2; 2; 0; 0; 2; 1; 1; 0; 1; 0]
 [1; 2; 1; 2; 0; 0; 1; 2; 2; 0]
 [2; 1; 2; 1; 1; 2; 2; 1; 2; 2]
 [1; 0; 1; 1; 0; 0; 2; 1; 0; 1]
 [0; 0; 0; 2; 0; 0; 2; 2; 1; 0]
 [1; 2; 1; 1; 0; 0; 2; 1; 1; 0]]

metameric images:
[[[0; 0; 0; 2; 0; 0; 0; 2; 1; 2]
 [0; 2; 0; 0; 0; 0; 1; 0; 1; 0]
 [0; 2; 1; 0; 2; 2; 0; 1; 0; 2]
 [0; 0; 2; 2; 0; 0; 2; 0; 2; 0]
 [2; 2; 0; 0; 2; 1; 1; 0; 1; 0]
 [1; 2; 1; 2; 0; 0; 1; 2; 2; 0]
 [2; 2; 1; 1; 2; 1; 2; 2; 1; 2]
 [1; 0; 1; 1; 0; 0; 2; 1; 0; 1]
 [0; 0; 0; 2; 0; 0; 2; 2; 1; 0]
 [1; 2; 1; 1; 0; 0; 2; 1; 1; 0]];
 [[0; 0; 0; 2; 0; 0; 0; 2; 1; 2]
 [0; 2; 0; 0; 0; 0; 1; 0; 1; 0]
 [0; 2; 1; 0; 2; 2; 0; 1; 0; 2]
 [0; 0; 2; 2; 0; 0; 2; 0; 2; 0]
 [2; 2; 0; 0; 2; 1; 1; 0; 1; 0]
 [1; 2; 1; 2; 0; 0; 1; 2; 2; 0]
 [2; 1; 2; 1; 1; 2; 2; 1; 2; 2]
 [1; 0; 1; 1; 0; 0; 2; 1; 0; 1]
 [0; 0; 0; 2; 0; 0; 2; 2; 1; 0]
 [1; 2; 1; 1; 0; 0; 2; 1; 1; 0]]]

Candidate images per iteration:
252 846 2820 3288 2274 7358 5103 4554 3393 14616 58320 28647 16956 5073
6003 22923 10458 4734 1530 516 804 2412 1068 1050 120 120 120 72 216
144 144 72 72 72 72 216 144 144 72 72 48 48 48 48 24 24 12 12 4 2 2 2
2 2 2 2 2 2 2 2 2 2 2
    
```

FIGURE 5 Reconstructing an image from its local histogram with $m = 3$. In this example, the solution set contains two images.

worst case scenario is reached when all histograms considered are pseudo-linear, in which case a maximum of

$$\frac{m^2!}{q!|A|^{-r}(q+1)!r} \left(\frac{m!}{p!|A|^{-s}(p+1)!s} \right)^{2(n-m)}$$

hypotheses must be considered. Examples of the number of hypotheses by the above equation for a small set of n , A , and m are given below

n	$ A $	m	#Hypotheses
10	2	2	393216
10	2	3	602654094
10	3	2	786432
10	3	3	131651795681280
11	2	2	1572864
11	2	3	5423886846
11	3	2	3145728
11	3	3	4739464644526080

Note: In practice, the number of hypotheses in memory is considerably smaller. Consider the case of $n = 3$ and $m = 2$.

The initial 3 histograms h_{11} , h_{21} , and h_{12} generates hypotheses for the $m^2 - 1 = 3$ values in $|X_{22} \cap (X_{11} \cup X_{21} \cup X_{12})|$, for which there are only different $\binom{3+|A|-1}{|A|-1}$ histograms, and h_{22} must be $A^{\binom{3+|A|-1}{|A|-1}}$ out of the possible $\binom{4+|A|-1}{|A|-1}$ histograms. Similarly, if the histogram difference contains zero-values, then any candidate, which has a non-zero value at the corresponding histogram point can be discarded. This happens often, when $m^2 < |A|$.

6. Conclusion

In this article, we have considered locally overlapping histograms of functions from discrete domains and codomains of any dimension. Histograms of signals and images have been studied in the literature extensively, and particularly, the seminal work on locally orderless images (Koenderink and Doorn, 1999), the notion of local histograms has gained a solid theoretical basis. The authors' work map discrete functions and histograms into the continuous domain, in a manner that

makes differentiation of discrete functions well-posed. Their work, however, left the essential question unanswered: what is the expression power of histograms? In Sporrying and Darkner (2022) a partial answer is given to this question for binary signals with densely overlapping histograms, and in this article, we extend this work for non-binary discrete functions of any dimension and with windows of any overlap, and in this article we have:

- Presented a simple algebra for histograms of discrete domain and co-domains based on non-overlapping sets.
- For a given set of covering sets in the domain, we have given a constructive method for identifying unique, non-overlapping sets which cover the domain.
- We have given an equation for the size of the set of all possible histograms based on the set of unique, non-overlapping domain sets.
- For a specific set of histograms of the individual unique, overlapping and non-overlapping sets, we have given
 - an equation for calculating the corresponding histograms of any set in the domain and
 - an equation for counting the total number of functions with these histograms.
- Presented an algorithm, which given a set of overlapping histograms, produce the set of functions, which share these histograms.

Understanding the expression power of local histograms is not done. For one, we still seek to connect the results obtained for discrete functions with the continuous domain.

References

- Gabor, D. (1946). Theory of communication. Part 1: the analysis of information. *J. Inst. Electr. Eng.* 93, 429–441. doi: 10.1049/ji-3-2.1946.0074
- Koenderink, J. J., and Doorn, A. J. V. (1999). The structure of locally orderless images. *Int. J. Comput. Vis.* 31, 159–168.
- Sporrying, J., and Darkner, S. (2022). Reconstructing binary signals from local histograms. *Entropy* 24, 433. doi: 10.3390/e24030433

Data availability statement

The code is publicly available at: <https://github.com/sporrying/reconstructionFromHistograms>.

Author contributions

JS and SD: conceptualization and writing—review and editing. JS: formal analysis, methodology, software, and writing—original draft. Both authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher. All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Stover, C. (2022). *Finite Additivity*. From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein. Available online at: <https://mathworld.wolfram.com/FiniteAdditivity.html>

Wu, Y. N., Zhu, S. C., and Liu, X. (2000). Equivalence of julesz ensembles and frame models. *Int. J. Comput. Vis.* 38, 247–265. doi: 10.1023/A:1008199424771