Approximate Testing of Visual Properties

Sofya Raskhodnikova[1]

MIT Laboratory for Computer Science, Cambridge MA 02139, USA,
sofya@theory.lcs.mit.edu,
WWW home page: http://theory.lcs.mit.edu/~sofya/

Abstract. We initiate a study of property testing as applied to visual properties of images. Property testing is a rapidly developing area investigating algorithms that, with a small number of local checks, distinguish objects satisfying a given property from objects which need to be modified significantly to satisfy the property. We study visual properties of discretized images represented by $n \times n$ matrices of binary pixel values. We obtain algorithms with query complexity independent of n for several basic properties: being a half-plane, connectedness and convexity.

1 Introduction

> We chose to investigate *connectedness* because of a belief that this predicate is nonlocal in some very deep sense; therefore it should present a serious challenge to any basically local, parallel type of computation.
>
> PERCEPTRONS
> *Marvin Minsky and Seymour Papert*

Images are typically so large that it might be expensive to read every single bit of them. It is natural to ask what properties of an image can be detected by *sublinear* algorithms that read only a small portion of the image. In general, most problems are not solvable exactly with that restriction. Property testing [16, 11] (see [15, 9] for surveys) is a notion of approximation tailored for decision problems and widely used for studying sublinear algorithms. Property tests distinguish inputs with a given property from those that are *far* from satisfying the property. *Far* means that many characters of the input must be changed before the property arises in it. The query complexity of a property test is the number of characters it reads. The goal is to design tests with sublinear complexity.

Image analysis is one area potentially well suited to the property testing paradigm. Some salient features of an image may be tested by examining only a small part thereof. Indeed, one motivation for this study is the observation that the eye focuses on relatively few places within an image during its analysis. The analogy is not perfect due to the eye's peripheral vision, but it suggests that property testing may give some insight into the visual system.

In this paper, we present tests for a few properties of images. All our tests have complexity independent of the image size, and therefore work well even for huge images. We use image representation popular in learning theory (see, e.g., [14, 13]). Each image is represented by an $n \times n$ matrix M of pixel values. We focus on black and white images given by binary matrices with black denoted by 1 and white denoted by 0. To keep the correspondence with the plane, we index the matrix by $\{0, 1, \ldots, n-1\}^2$, with the lower left corner being $(0, 0)$ and the upper left corner being $(0, n-1)$. The object is a subset of $\{0, 1, \ldots, n-1\}^2$ corresponding to black pixels; namely, $\{(i, j) | M_{i,j} = 1\}$.

1.1 Property Testing in the Pixel Model

The *distance* between two images of the same size is defined as the number of pixels (matrix entries) on which they differ. (Two matrices of different size are considered to have infinite distance.) The *relative distance* is the ratio of the distance and the number of pixels in the image. A *property* is defined as a collection of pixel matrices. The distance of an image (matrix) M to a property $\mathcal{P}$ is $\min_{M' \in \mathcal{P}} dist(M, M')$. Its *relative distance* to $\mathcal{P}$ is its distance to $\mathcal{P}$ divided by the size of the image matrix. An image is ε-far from $\mathcal{P}$ if its relative distance to $\mathcal{P}$ is at least ε. If the image is not ε-far from $\mathcal{P}$, it is ε-close to it.

A property is (ε, q)*-testable* if there is a randomized algorithm that for every input matrix M queries at most q entries of M and with probability at least $\frac{2}{3}$ distinguishes between matrices with the property and matrices which are ε-far from having it. The algorithm is referred to as an (ε, q)*-test.* This definition allows tests to have *2-sided error.* An algorithm has *1-sided error* if it always accepts an input that has the property.

1.2 Our Results

We present tests for three visual properties: being a half-plane, convexity and connectedness. The number of queries in all tests is independent of the size of the input. The algorithm for testing if the input is a half-plane is a 1-sided error test with $\frac{2 \ln 3}{\varepsilon} + o(\frac{1}{\varepsilon})$ queries. The convexity test has 2-sided error and makes $O(1/\varepsilon^2)$ queries. Finally, the connectedness test has 1-sided error and makes $O\left(\frac{1}{\varepsilon^2} \log^2 \frac{1}{\varepsilon}\right)$ queries.

1.3 Related Results in Property Testing

Previous papers on property testing in computational geometry [7, 6] consider a model different from ours, where the input is the set of object points and a query i produces coordinates of the ith point. Their results, in general, are incomparable to ours. In their model, the problems we consider would have query complexity dependent on the number of points in the object. But they are able to study properties which are trivially testable in our model because all instances are either close to having the property or close to not having it. An example is the property that a given graph is a Euclidean minimum spanning tree of a given point set in the plane [7].

Another related work is [10] which studies properties of d-dimensional matrices. It gives a class of properties which are testable with a number of queries polynomial in $1/\varepsilon$. It does not seem applicable to our geometric properties.

Goldreich and Ron [12] study property testing in bounded degree graphs represented by adjacency lists. Note that an image in the pixel model can be viewed as a graph of degree 4 where vertices correspond to black pixels and they are connected by an edge if the corresponding entries in the image matrix are adjacent. (See the definition of the *image graph* in the beginning of section 4.) Goldreich and Ron measure distance between graphs as the ratio of the number of edges that need to be changed to transform one graph into the other over the maximum possible number of edges in the graphs with the given number of vertices and degree. In our case, the distance between two image graphs corresponds to the fraction of points (vertices) on which they differ, i.e. the edge structure of the graphs is fixed, and only vertices can be added or removed to transform one graph into another. Our connectedness test is exactly the same as the connectivity test in [12], with one minor variation due to different input representation and the fact that the pixel model allows graphs with a small number of vertices. (In the bounded degree graph model, the number of vertices is a part of the input.) However, since our distance measures are different, their proof of correctness of the algorithm does not apply to the pixel model.

One more paper that studies fast algorithms for connectedness in graphs is [5]. It shows how to approximate the number of connected components in an arbitrary graph in a sublinear time.

1.4 Related Results in Learning

In property testing terminology, a PAC (probably approximately correct) learning algorithm [17] is given oracle access (or access via random samples) to an unknown *target* object with the property $\mathcal{P}$ and has to output a *hypothesis* which is within relative distance ε to the target with high probability. If the hypothesis is required to have the property $\mathcal{P}$, the learning algorithm is *proper*. As proved in [11], a proper PAC learning algorithm for $\mathcal{P}$ with sampling complexity $q(\varepsilon)$ implies a (2-sided error) $(\varepsilon, q(\varepsilon/2) + O(1/\varepsilon))$-test for $\mathcal{P}$.

Learning half-planes exactly is considered in [14]. This work gives matching upper and lower bound of $\Theta(\log n)$ for the problem. In the PAC model, a proper learning algorithm with $O(1/\varepsilon \log(1/\varepsilon))$ sampling complexity follows from [3]. Together with the [11] result above, it implies a (2-sided error) $(\varepsilon, O(1/\varepsilon \log(1/\varepsilon)))$-test for the half-plane property. Our result for testing half-planes is a modest improvement of shaving off the log factor and making the error 1-sided.

The generic approach of [11] for transforming PAC proper learners into property tests does not seem to work well for convexity and connectedness. The complexity of PAC learning algorithms is at least proportional to Vapnik Chervonenkis (VC) dimension[1] [8]. Since VC dimension of convexity is $\Theta(n)$ and VC dimension of connectedness is $\Theta(n^2)$, the corresponding tests obtained by the generic approach have query complexity guarantee $O(n)$ and $O(n^2)$, respectively. Our tests for these properties have query complexity independent of n.

2 Testing if an Image Is a Half-Plane

First we present an algorithm for testing whether the image is a half-plane. An image is a *half-plane* if there is a vector $w \in \mathbb{R}^2$ and a number $a \in \mathbb{R}$ such that a pixel x is black if and only if $w^T x \geq a$. The algorithm first finds a small region within which the dividing line falls. Then it checks if pixels on one side of the region are white and on the other side are black.

Call pixels $(0,0), (0,n-1), (n-1,0), (n-1,n-1)$ *corners*. Call the first and the last row and the first and the last column of the matrix *sides*. For a pair of pixels p_1, p_2, let $\ell(p_1, p_2)$ denote the line[2] through p_1, p_2. Let $R_1(p_1, p_2)$ and $R_2(p_1, p_2)$ denote the regions into which $\ell(p_1, p_2)$ partitions the image pixels not on the line.

HALF-PLANE TEST $T_1(\varepsilon)$

Given access to an $n \times n$ pixel matrix,

1. Query the four corners. Let s be the number of sides with differently colored corners.
   (a) If $s = 0$ (all corners are of the same color c), query $\frac{\ln 3}{\varepsilon}$ pixels independently at random. Accept if all of them have color c. Reject otherwise.
   (b) If $s = 2$,
      i. For both sides with differently colored corners, do binary search of pixels on the side to find two differently colored pixels within distance less than $\varepsilon n/2$. For one side, call the white pixel w_1 and the black pixel b_1. Similarly, define w_2 and b_2 for the second side.
      ii. Let $W_i = R_i(w_1, w_2)$ and $B_i = R_i(b_1, b_2)$ for $i = 1, 2$. W.l.o.g., suppose W_2 and B_1 intersect while W_1 and B_2 do not. Query $\frac{2\ln 3}{\varepsilon}$ pixels from $W_1 \cup B_2$ independently at random. Accept if all pixels from W_1 are white, all pixels from B_2 are black. Otherwise, reject.
   (c) If s is not 0 or 2, reject.

---

[1] The *VC dimension* is the cardinality of the largest set $X \subseteq \{0, \ldots, n-1\}^2$ shattered by $\mathcal{P}$. A set $X \subseteq \{0, \ldots, n-1\}^2$ is *shattered* by $\mathcal{P}$ if for every partition (X_0, X_1) of X, $\mathcal{P}$ contains a matrix M with $M_x = 1$ for all $x \in X_1$ and $M_x = 0$ for all $x \in X_0$.

[2] Whenever a geometric notion (e.g., line, angle, convex hull) is used without a definition, it refers to the standard continuous notion. All discretized notions are defined.

Theorem 1. *Algorithm T_1 is a 1-sided error $(\varepsilon, \frac{2\ln 3}{\varepsilon} + o(\frac{1}{\varepsilon}))$-test for the half-plane property.*

Proof. The algorithm queries at most $\frac{2\ln 3}{\varepsilon} + O(\log(1/\varepsilon))$ pixels. To prove correctness, we need to show that all half-planes are always accepted, and all images that are ε-far from being half-planes are rejected with probability at least $2/3$.

Case (a) [0 differently colored sides]: The image is a half-plane if and only if it is unicolored. If it is unicolored, the test always accepts since it never finds pixels of different colors. If the image is ε-far from being a half-plane, it has at least εn^2 pixels of a wrong color. Otherwise, it can be made unicolored, and hence a half-plane, by changing less than an ε-fraction of pixels. The test fails to find an incorrectly colored pixel and accepts with probability at most $(1-\varepsilon)^{\ln 3/\varepsilon} < 1/3$.

Case (b) [2 differently colored sides]: The test always accepts all half-planes because it rejects only if it finds two white pixels and two black pixels such that the line through the white pixels intersects the line through the black pixels.

It remains to show that if an image is ε-far from being a half-plane, it is rejected with probability $\geq 2/3$. We prove the contrapositive, namely, that if an image is rejected with probability $< 2/3$, modifying an ε fraction of pixels can change it into a half-plane.

Suppose that an image is accepted with probability $\geq 1/3 = e^{-\ln 3} > (1-\varepsilon/2)^{2\ln 3/\varepsilon}$. That means that $< \varepsilon/2$ fraction of pixels from which we sample in step 1(b)ii differ from the color of their region (white for W_1 and black for B_2). Note also that there are at most $\varepsilon n/2$ pixels outside of $W_1 \cup B_2$. Changing the color of all black pixels in W_1 and all white pixels in B_2 and making all pixels outside of those regions white, creates a half-plane by changing $< \varepsilon$ fraction of the pixels, as required.

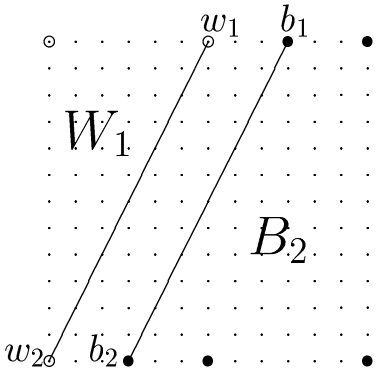


Fig. 1. Half-plane test

Case (c) [everything else]: The number of image sides with differently colored corners is even (0, 2, or 4). That holds because the cycle $((0,0),(n-1,0),(n-1,n-1),(0,n-1),(0,0))$ visits a vertex of a different color every time it moves along such a side. So, the only remaining case is 4 differently colored sides. In this case, the image cannot be a half-plane. The test always rejects. □

3 Convexity Testing

The image is *convex* if the convex hull of black pixels contains only black pixels. The test for convexity first roughly determines the object by querying pixels on the $n/u \times n/u$ grid with a side of size $u = \Theta(\varepsilon n)$. Then it checks if the object corresponds to the rough picture it obtained.

For all indices i, j divisible by u, call the set $\{(i', j') | \, i' \in [i, i{+}u], j' \in [j, j{+}u]\}$ a *u-square*. We refer to pixels $(i, j), (i + u, j)(i + u, j + u)$, and $(i, j + u)$ as its corners.

Convexity test $T_2(\varepsilon)$
Given access to an $n \times n$ pixel matrix, 1. Query all pixels with both coordinates divisible by $u = \lfloor \varepsilon n/120 \rfloor$. 2. Let B be the convex hull of discovered black pixels. Query $\frac{5}{\varepsilon}$ pixels from B independently at random. Reject if a white pixel in B is found in steps 1 or 2. 3. Let W be the union of all u-squares which contain no pixels from B. Query $\frac{5}{\varepsilon}$ pixels from W independently at random. Reject if a black pixel is found. Otherwise accept.

Lemma 1, used in the analysis of the convexity test, asserts that the number of pixels outside $B \cup W$ is small.

Lemma 1. *In an $n \times n$ image, let B be the convex hull of black pixels with coordinates divisible by u. Let W be the union of u-squares which contain no pixels from B. Let the "fence" F be the set of pixels not contained in B or W. Then F contains at most $4un$ pixels.*

Proof. Intuitively, F is the largest when it contains all u-squares along the sides of the image. We call u-squares that are not fully contained in B or W *fence u-squares.* Note that F is covered by *fence u-squares.* Therefore, to prove the lemma it is enough to show that there are at most $4n/u$ *fence u-squares.*

To count the *fence u-squares*, we define a cyclic ordering on them. To do that, we describe a walk that connects centers of all *fence u-squares.* The walk goes from one center to the next by traveling left, right, up or down. It visits the centers of *fence u-squares* by traveling clockwise and keeping the boundary between F and W on the left-hand side. Each *fence u-square* is visited because it intersects with some u-square in W in at least one pixel.

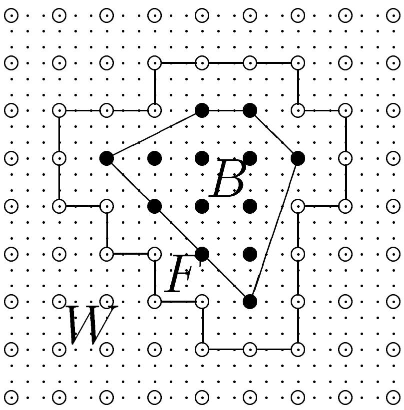


Fig. 2. Convexity test

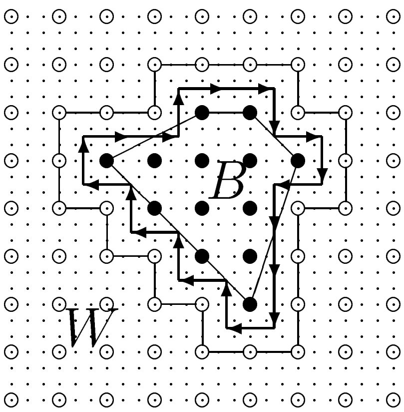
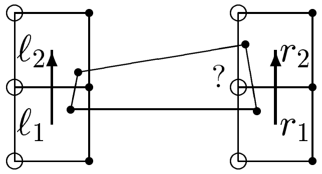


Fig. 3. Walk over fence u-squares

There are n/u rows of u-squares. We claim that from each of these rows the walk can travel up at most once. Suppose for contradiction that it goes up twice, from ℓ_1 to ℓ_2 and from r_1 to r_2, where ℓ_1 and r_1 are *fence u-squares* with centers in row $(k+0.5)u$, and ℓ_2 and r_2 are *fence u-squares* with centers in row $(k+1.5)u$ for some integer k.



W.l.o.g. suppose that the centers of l_1, l_2 are in a column with a lower index than the centers of r_1, r_2. Since the walk keeps the boundary between W and F on the left-hand side, the left corners of ℓ_1, ℓ_2, r_1, r_2 are in W. By definition of *fence u-squares*, ℓ_1, ℓ_2, r_1, r_2 each contain a pixel from B. The common left corner of r_1 and r_2 is also in B, since B is convex. But this is a contradiction because W and B are disjoint.

Thus, the walk can travel up only once per row. Similarly, it can travel down only once per row, and travel left (right) only once per column. Since there are n/u rows (columns) of u-squares, the walk can have at most $4n/u$ steps. As it visits all *fence u-squares*, there are at most $4n/u$ of them. Since each u-square contributes u^2 pixels, the number of pixels in F is at most $4nu$. $\square$

The analysis of the convexity test uses the fact that if an image is convex, W contains only a small number of black pixels. Proposition 1 proves this fact for a special case of an image which is "invisible" on the big grid. Later, we use the proposition to handle the general case in lemma 2.

Proposition 1. *In an $n \times n$* convex *image, if all pixels with both coordinates divisible by u are white, then the image contains less than $2un$ black pixels.*

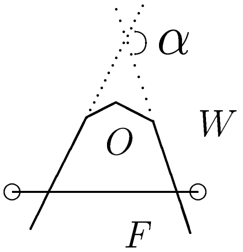
Proof. Let $black(r)$ denote the number of black pixels in a row r. If each row contains fewer than $u-1$ pixels, the total number of black pixels is at most un. Otherwise, consider a row r with $black(r) \geq u$. Let integers k and t be such that $r = ku + t$ and $0 \leq t < u$. Since the image is convex, black pixels of every fixed row must have consecutive column indices. Since every pixel with both coordinates divisible by u is white, $black(ku) < u$ and $black((k+1)u) < u$.

Because of the convexity of the object, if $black(r_1) < black(r)$ for a row $r_1 > r$ then $black(r_2) \leq black(r_1)$ for all rows $r_2 > r_1$. Similarly, if $black(r_1) < black(r)$ for a row $r_1 < r$ then $black(r_2) \leq black(r_1)$ for all rows $r_2 < r_1$. Thus, all rows r_2 excluding $ku+1, ku+2, \ldots, (k+1)u-1$ have $black(r_2) < u$. Together, they contain $< (n-u)u$ black pixels. Cumulatively, the remaining $u-1$ rows contain $< (u-1)n$ pixels. Therefore, the image contains less than $2un$ black pixels. $\square$

Lemma 2. *In an $n \times n$* convex *image, let W be the union of all u-squares which contain no pixels from B. Then W contains less than $8un$ black pixels.*

Proof. As before, let F be the set of all pixels not contained in B or W. We call pixels on the boundary between F and W with both coordinates divisible by u *fence posts*. Since all *fence posts* are white, any portion of the object protruding into W has to squeeze between the *fence posts*. We show that there are at most three large protruding pieces, each of which, by proposition 1, contains less than

$2un$ pixels. All other sticking out portions fall close to the fence and are covered by the area containing less than $2un$ pixels.



Let O be the boundary of our convex object. O can be viewed as a piecewise linear trajectory on the plane that turns 360°. Whenever O leaves region F to go into W, it has to travel between two *fence posts*. Whenever O comes back into F, it has to return between the same fence posts because the object is convex and *fence posts* do not belong to it. The figure depicts an excursion of O into W with accumulated turn α.

Notice that since O turns 360° total, at most 3 excursion into W have accumulated turn $> 90°$. Each of them can be viewed as delineating a part of our convex object, cut off by the line between the *fence posts*. This part of the object is convex, and therefore, by proposition 1, has $< 2un$ pixels. This gives us a total of $< 6un$ pixels for the protruding parts where O turns more than 90°.

Consider any excursion into W where O leaves F between *fence posts* p_1 and p_2 and turns $\leq 90°$ before coming back. Any such trajectory part lies inside the circle of diameter u containing p_1 and p_2. The half of the circle protruding into W is covered by a half of a u-square. By an argument identical to counting *fence squares* in lemma 1, there are at most $4n/u$ segments of the F/W boundary between adjacent fence posts. Therefore, the total number of pixels that might be touched by the parts of the object, described by O's excursions into W that turn $\leq 90°$ is at most $4n/u \cdot u^2/2 = 2un$.

Thus, the total number of black pixels in W is at less than $8un$. □

Theorem 2. *Algorithm T_2 is a $(\varepsilon, O(1/\varepsilon^2))$-test for convexity.*

Proof. The test makes $(n/u)^2 + O(1/\varepsilon) = O(1/\varepsilon^2)$ queries. We bound failure probability, considering convex and far from convex images separately.

If the input image is convex, B contains only black pixels. The test never rejects in step 2. By lemma 2, the fraction of black pixels in W is $< 8u/n = \varepsilon/15$. By the union bound, the probability that the test rejects in step 3 is $< \frac{\varepsilon}{15}\frac{5}{\varepsilon} = \frac{1}{3}$.

If the input image is ε-far from convex, it has $\geq 2\varepsilon n^2/5$ white pixels in B or $\geq 2\varepsilon n^2/5$ black pixels in W. Otherwise, we could make the image convex by making all pixels in W white and all remaining pixels black. It would require $< 2\varepsilon n^2/5$ changes in B, $< 2\varepsilon n^2/5$ changes in W, and by lemma 1, $\leq 4un < \varepsilon n^2/5$ changes in F. Thus, the distance of the image to convex would be less than εn^2.

Suppose w.l.o.g. that there are $\geq 2\varepsilon/5$ black pixels in W. Step 3 will fail to find a black pixel with probability $\leq (1 - \frac{2\varepsilon}{5})^{5/\varepsilon} \leq e^{-2} < \frac{1}{3}$. □

4 Connectedness Testing

Define the *image graph* $G_M = (V, E)$ of image matrix M by $V = \{(i,j) | M_{i,j} = 1\}$ and $E = \{((i_1, j), (i_2, j)) | \ |i_1 - i_2| = 1\} \cup \{((i, j_1), (i, j_2)) | \ |j_1 - j_2| = 1\}$. In other words, the image graph consists of black pixels connected by the grid lines. The image is *connected* if its image graph is connected. When we say that the image has k connected components, we are also referring to its image graph.

The test for connectedness looks for isolated components of size less than $d = 4/\varepsilon^2$. We prove that a significant fraction of pixels are in such components if the image is far from connected. When a small isolated component is discovered, the test rejects if it finds a black pixel outside of the component. Lemma 3 implies that if an image is far from connected, it has a large number of connected components. An averaging argument in lemma 4 demonstrates that many of them have to be small. This gives rise to a simple test T_3, which is later improved to test T_4 with more careful accounting in proposition 2.

Both tests for connectedness and proposition 2 are adopted from [12]. The only change in the tests, besides parameters, is that after finding a small component, we make sure there is some point outside of it before concluding that the image is far from connected.

<u>CONNECTEDNESS TEST $T_3(\varepsilon)$</u>

Let $\delta = \frac{\varepsilon^2}{4} - o(1)$ and $d = 4/\varepsilon^2$. Given access to an $n \times n$ pixel matrix,

1. Query $2/\delta$ pixels independently at random.
2. For every pixel (i, j) queried in step 1, perform a breadth first search (BFS) of the image graph starting from (i, j) until d black pixels are discovered or no more new black pixels can be reached; i.e., for each discovered black pixel query all its neighbors if they haven't been queried yet. If no more new black pixels can be reached, a small connected component has been found.
3. If a small connected component is discovered for some (i, j) in step 2, query $2/\varepsilon$ pixels outside of the square $[i - d, i + d] \times [j - d, j + d]$ independently at random. If a black pixel is discovered, reject. Otherwise (if no small connected component is found or if no black pixel is discovered outside of the small component), accept.

Lemma 3. *If an $n \times n$ image contains at most p connected components, they can be linked into one connected component by changing at most $n(\sqrt{2p} + O(1))$ pixel values from white to black.*

Proof. Let $s = n\sqrt{2/p}$. To turn the image into one connected component, we first add the comb-like set $S = \{(i, j) |\, j = n-1 \text{ or } i = n-1 \text{ or } s \text{ divides } i\}$. Now every connected component is linked to S by adding at most $s/2$ pixels leading to the nearest "tooth of the comb". That is, if a component contains a pixel $(ks+\ell, j)$ for an integer k and $0 \le \ell \le s/2$, add pixels $(ks+1, j), (ks+2, j), \ldots, (ks+\ell-1, j)$. Otherwise (a component contains a pixel $(ks+\ell, j)$ for integer k and $s/2 < \ell < s$), add pixels $(ks + \ell + 1, j), (ks + \ell + 2, j), \ldots, (ks + s - 1, j)$. The first stage adds $|S| = n(n/s + O(1))$ pixels and the second, less than $s/2$ per connected component, adding the total of $n(n/s + O(1)) + ps/2 = n\sqrt{2p} + O(1)$ pixels. □

Lemma 4. *If an image is ε-far from connected, at least an $\frac{\varepsilon^2}{4} - o(1)$ fraction of its pixels are in connected components of size less than $d = 4/\varepsilon^2 + o(1)$.*

Proof. Consider an $n \times n$ ε-far from connected image with p connected components. By lemma 3, changing $\leq n(\sqrt{2p} + O(1))$ pixels makes it connected. Then $n(\sqrt{2p} + O(1)) \geq \varepsilon n^2$, and $p \geq \varepsilon^2 n^2/2 - O(n)$. Let b be the number of black pixels. The average component size is $b/p \leq n^2/(\varepsilon^2 n^2/2 - O(n)) = 2/\varepsilon^2 + o(1)$. Thus, the fraction of components of size up to $d = \frac{4}{\varepsilon^2} + o(1)$ is $\geq 1/2$. That is, there are $\geq p/2 = \varepsilon^2 n^2/4 - O(n)$ such components. Since each connected component contains a pixel, $\geq \varepsilon^2/4 - o(1)$ fraction of pixels are in connected components of size d. □

Theorem 3. *Algorithm T_3 is a 1-sided $(\varepsilon, O(\varepsilon^{-4}))$-test for connectedness.*

Proof. The algorithm accepts all connected images because it rejects only if an isolated component and some pixel outside of it are found.

It remains to show that an ε-far from connected image is rejected with probability at least $2/3$. By lemma 4, such an image has at least a δ fraction of its pixels in connected components of size less than d. The probability that step 1 fails to find a pixel from a small connected component is $(1-\delta)^{2/\delta} \leq e^{-2}$. In step 2, $3d - 1$ queries are sufficient to discover that a component of size $d - 1$ is isolated because it has at most $2d$ neighboring white pixels. There are at least $\varepsilon n^2 - 4d^2$ black pixels outside of the $2d \times 2d$ square containing the small isolated component. Step 3 will fail to find a black pixel with probability $(1-\varepsilon)^{2\varepsilon} \leq e^{-2}$. By the union bound, the failure probability is at most $2/e^2 < 1/3$.

The number of queries is at most $2/\delta \times (3d - 1) + 2/\varepsilon = O(\varepsilon^{-4})$. □

The algorithm can be improved by employing the Goldreich-Ron trick [12] of considering small components of different sizes separately. The following proposition is adopted from [12].

Proposition 2. *If an image has at least C connected components of size less than d, there is $\ell \leq \log d$ such that at least $\frac{C \cdot 2^{\ell-1}}{\log d}$ points are in connected components of size between $2^{\ell-1}$ and $2^\ell - 1$.*

Proof. For some $\ell \leq \log d$, the image has at least $C/\log d$ connected components of size between $2^{\ell-1}$ and $2^\ell - 1$. Each of them contains at least $2^{\ell-1}$ points. □

<u>(IMPROVED) CONNECTEDNESS TEST $T_4(\varepsilon)$</u>

Let $\delta = \frac{\varepsilon^2}{4} - o(1)$ and $d = 4/\varepsilon^2$. Given access to an $n \times n$ pixel matrix,

1. For $\ell = 1$ to $\log d$
   (a) Query $\frac{4 \log d}{\delta 2^\ell}$ pixels independently at random.
   (b) For every pixel (i, j) queried in step 1a, perform a BFS of the image graph starting from (i, j) until 2^ℓ black pixels are discovered or no more new black pixels can be reached (a small connected component has been found).
2. If a small connected component is discovered for some (i, j) in step 1, proceed as in step 3 of algorithm T_3.

Theorem 4. *Algorithm T_4 is a 1-sided $\left(\varepsilon, O\left(\frac{1}{\varepsilon^2}\log^2\frac{1}{\varepsilon}\right)\right)$-test for connectedness.*

Proof. The algorithm accepts all connected images because it rejects only if an isolated component and some pixel outside of it are found.

If an $n \times n$ image is ε-far from connected, by the proof of lemma 4, it has at least a δn^2 connected components of size less than d. Proposition 2 implies that for some $\ell < \log d$, at least an $\frac{\delta \cdot 2^{\ell-1}}{\log d}$ fraction of its points are in connected components of size between $2^{\ell-1}$ and $2^\ell - 1$. For this ℓ, the probability that step 1 fails to find a pixel from a component of size between $2^{\ell-1}$ and $2^\ell - 1$ is at most e^{-2}. The rest of the correctness analysis is the same as in theorem 3.

The number of queries is at most $\log d \cdot O\left(\frac{\log d}{\delta}\right) + 2/\varepsilon = O\left(\frac{1}{\varepsilon^2}\log^2\frac{1}{\varepsilon}\right)$. □

5 Conclusion and Open Problems

Employing the Paradigm from the Half-plane test The strategy employed in the half-plane test of section 2 is very simple. First we approximately learn the position of the dividing line. Then, using the fact that all half-planes consistent with our knowledge of the dividing line differ only on a fixed $\varepsilon/2$ fraction of the pixels, we randomly check if the matrix corresponds to these half-planes on the remaining pixels.

This suggests a general paradigm for transforming PAC learning algorithms into property tests with *1-sided error*. Namely, consider a property $\mathcal{P}$ where all objects with $\mathcal{P}$ which are $\varepsilon/2$-close to a given object are the same on all but $\varepsilon/2$ fraction of the points. In addition, assume there is a proper PAC learning algorithm with sampling complexity $q(n, \varepsilon)$. Then the following test for $\mathcal{P}$ has 1-sided error and query complexity $q(n, \varepsilon/2) + O(1/\varepsilon)$: learn the property within relative error of $\varepsilon/2$ and then randomly test the object on points where all objects $\varepsilon/2$-close to the hypothesis coincide. The proof of this fact is very similar to the case 2 of the analysis of the half-plane test.

Extensions and Lower Bounds We restricted our attention to images representable by binary matrices. However, in real life images have many colors (or intensity values). Property tests for images represented by integer-valued matrices would be a natural generalization. For example, one can generalize convexity in the following way. Call an image represented by an $n \times n$ matrix with values in R *convex* if the corresponding function $\{0, 1, \ldots, n-1\}^2 \to R$ is convex.

A straightforward extension of our tests to d dimensions seems to give tests with dependence on d, and thus dependent on the size of the image. It would be interesting to investigate if this dependence is necessary.

It is known that testing some properties requires a number of queries linear in the size of the input [4, 2]. However, known hard properties do not seem to have a natural geometric interpretation. It would be nice to find natural 2-dimensional visual properties which are hard to test. One such result follows directly from [1], which shows that testing whether a string of length n is a shift of another string requires $\Omega(n^{1/2})$ queries. This implies that testing whether the lower half of an $n \times n$ image is a shift of the upper half requires $\Omega(n^{1/2})$ queries. It would be interesting to find even harder visual properties.

Acknowledgements The author would like to thank Michael Sipser for proposing the problem and many useful discussions. She is also very grateful to Piotr Indyk for help and moral support.

References

1. T. Batu, F. Ergun, J. Kilian, A. Magen, S. Raskhodnikova, R. Rubinfeld, and R. Sami, A Sublinear Algorithm for Weakly Approximating Edit Distance, *Proceedings of the* 35^{th} *ACM STOC* (2003)
2. E. Ben-Sasson, P. Harsha, and S. Raskhodnikova, 3CNF Properties are Hard to Test, *Proceedings of the* 35^{th} *ACM STOC* (2003)
3. A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth, Learnability and the Vapnik-Chervonenkis dimension, *Journal of the Association for computing machinery* **36(4)** (1989) 929–965
4. A. Bogdanov, K. Obata, L. Trevisan, A linear lower bound on the query complexity of property testing algorithms for 3-coloring in bounded-degree graphs, *Proceedings of the* 42^{nd} *IEEE FOCS* (2002)
5. B. Chazelle, R. Rubinfeld, and L. Trevisan, Approximating the minimum spanning tree weight in sublinear time, *Proceedings of ICALP* (2001)
6. A. Czumaj and C. Sohler, Property testing with geometric queries, *Proceedings of the 9th European Symposium on Algorithms* (2001) 266–277
7. A. Czumaj, C. Sohler and M. Ziegler, Property testing in computational geometry, *Proceedings of the 8th European Symposium on Algorithms* (2000) 155–166
8. A. Ehrenfeucht, D. Haussler, M. Kearns and L. Valiant, A General Lower Bound on the Number of Examples Needed for Learning, *Information and Computation* **82(3)** (1989) 247–261
9. E. Fischer, The art of uninformed decisions: A primer to property testing, *The Computational Complexity Column of The Bulletin of the European Association for Theoretical Computer Science* **75** (2001) 97–126
10. E. Fischer and I. Newman, Testing of matrix properties, *Proceedings of the* 33^{rd} *ACM STOC* (2001) 286–295
11. O. Goldreich, S. Goldwasser and D. Ron, Property testing and its connection to learning and approximation, *Journal of the ACM* 45 (1998) 653–750
12. O. Goldreich and D. Ron, Property Testing in Bounded Degree Graphs, *Proceedings of the* 28^{th} *ACM STOC* (1997)
13. E. Kushilevitz and D. Roth, On Learning Visual Concepts and DNF Formulae, *Machine Learning* (1996)
14. W. Maass and G. Turan, On the complexity of learning from counterexamples, *Proceedings of the* 30^{th} *IEEE FOCS* (1989) 262–267
15. D. Ron, Property testing (a tutorial), In *Handbook of Randomized Computing* (S. Rajasekaran, P. M. Pardalos, J. H. Reif and J. D. P. Rolimeds), Kluwer Press (2001)
16. R. Rubinfeld and M. Sudan, Robust characterization of polynomials with applications to program testing, *SIAM Journal of Computing* **25** (1996) 252–271
17. L. Valiant, A theory of the learnable, *Communications of the ACM* **27** (1984) 1134–1142