

An Integrated Fast Hough Transform for Multidimensional Data

Yanhui Li and Xiangchao Gan , Member, IEEE

Abstract—Line, plane and hyperplane detection in multidimensional data has many applications in computer vision and artificial intelligence. We propose Integrated Fast Hough Transform (IFHT), a highly-efficient multidimensional Hough transform algorithm based on a new mathematical model. The parameter space of IFHT can be represented with a single k -tree to support hierarchical storage and “coarse-to-fine” search strategy. IFHT essentially changes the least square data-fitting in Li’s Fast Hough transform (FHT) to the total least squares data-fitting, in which observational errors across all dimensions are taken into account, thus more practical and more resistant to data noise. It has practically resolved the problem of decreased precision of FHT for target objects mapped to boundaries between accumulators in the parameter space. In addition, it enables a straightforward visualization of the parameter space which not only provides intuitive insight on the number of objects in the data, but also helps with tuning the parameters and combining multiple instances if needed. In all simulated data with different levels of noise and parameters, IFHT surpasses Li’s Fast Hough transform in terms of robustness and precision significantly.

Index Terms—Hough Transform, multidimensional data, hyperplane detection, parameter space.

I. INTRODUCTION

LINE and plane detection has played a fundamental role in automatic object recognition and has many applications in computer vision and artificial intelligence. Since Hough earned a patent in 1962 for his method that allows efficient detection of lines in images [1], which we now call Hough transform (HT), more than 2500 papers have been devoted to its variants. HT has since been applied to various fields, such as image analysis [2], [3], auto-piloting [4], artificial intelligence [5] and data clustering [6], [7].

Most variants of Hough transform deal with 2D data and detect straight lines [8] and sometimes other curves with an adaption of the mathematic model [9], [10]. With the increasing

deployment of Light Detection and Ranging (LiDAR) equipment in vehicles and satellites [11], [12], Hough transform for 3D point clouds attracts more and more attention in auto-piloting and remote sensing research [13], [14].

The computational cost of Hough transform usually increases dramatically when it is extended to high dimensional data due to the “curse of dimensionality” [15]. Among many variants of Hough transform, Li’s Fast Hough transform (FHT) is one of the best that can efficiently deal with high dimensional data [16]. We noticed literatures in the field [17] sometimes refer Fast Hough transform to Brady’s radon transform-based algorithm [18]. To avoid confusion, we hereafter refer FHT to Li’s algorithm unless stated otherwise. Li’s FHT algorithm is highly efficient in terms of computation and storage in high-dimensional data. This is achieved mainly through a “coarse-to-fine” strategy. For n -dimension data, FHT transforms input data points into target objects (lines in 2D, planes in 3D, hyperplanes in higher dimension) in the parameter space and searches for intersection. It subsequently divides the parameter space into n subspaces, each represented by a k -tree. Each subspace is divided recursively into hypercubes from low to high resolution and further analysis is performed only on the hypercubes with votes exceeding a certain threshold. A hypercube receives a vote only when a hyperplane, corresponding to an input data point in the data space, intersects it. This hierarchical approach leads to a significant reduction on computational load and storage. The algorithm has wide applications [19] and has been implemented by some open source image analysis libraries, such as GANDALF [20].

In an n -dimensional dataset, FHT divides the parameter space into n independent subspaces. For example, a straight line in a 2D space can be modelled as $y = mx + c$, with $m \rightarrow \infty$ to describe vertical lines. A direct search of parameter space $[m, c]$ would be difficult if not impossible. FHT models straight lines with two equations $y = mx + c$, $|m| \leq 1$ and $x = m'y + c'$, $|m'| < 1$. This effectively divides the search space into two parameter subspaces (m, c) and (m', c') , with $|m| \leq 1$, $|m'| < 1$. Since m , c , m' and c' are all limited, efficient strategies can be designed to exploit all search space. The same model has been used in a plenary of other HT variants such as adaptive Hough transform [21].

Searching in multiple independent parameter subspaces poses several serious challenges. Though in theory any line would not appear in both parameter subspaces, e.g., $y = mx + c$, $|m| \leq 1$ and $x = m'y + c'$, $|m'| < 1$, in practice lines with the slope close to 1 need to be pursuit in both parameter subspaces when noise is considered. Thus, the n parameter subspaces in FHT must overlap marginally with each other to avoid potential gaps.

Manuscript received 14 January 2022; revised 3 April 2023; accepted 12 April 2023. Date of publication 21 April 2023; date of current version 4 August 2023. This work was supported by the National Key R&D Program of China under Grant 2022ZD0115703, in part by the National Natural Science Foundation of China under Grant 32170647, and in part by the National Science Foundation of Jiangsu Province in China under Grants BK20212010 and BE2022383, and in part by the Jiangsu Collaborative Innovation Center for Modern Crop Production. Recommended for acceptance by V. Patel. (Corresponding author: Xiangchao Gan.)

The authors are with the State Key Laboratory of Crop Genetics and Germplasm Enhancement and Utilization, Bioinformatics Center, Academy for Advanced Interdisciplinary Studies, Nanjing Agricultural University, Nanjing, Jiangsu 210095, China (e-mail: 2021201003@njau.edu.cn; gan@njau.edu.cn).

Digital Object Identifier 10.1109/TPAMI.2023.3269202

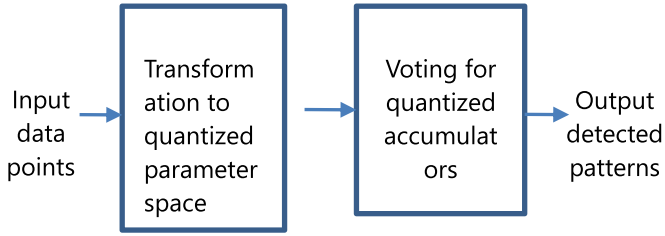


Fig. 1. Block diagram of Hough transform for line, plane and hyperplane detection.

With the increase of the data dimension, significant computational and storage costs are needed for the overlapped regions in the search subspaces.

To make matters worse, FHT operates in n independent parameter subspaces and each parameter subspace uses a unique transformation. As a consequence, testing whether objects detected in different parameter subspaces are from the same object in the original data space is a necessary and very challenging task and needs complicated algorithms. This was indeed noticed in both FHT patent and the original FHT publication and is listed as future work. However, to our best knowledge, there is still a lack of reliable method to resolve it after the initial publication of FHT decades ago.

In the paper, we propose an integrated fast Hough transform (IFHT) parameter space to model lines, planes and hyperplanes. IFHT allows us to represent the parameter space with a single k -tree. Apart from the obvious advantage on computation and storage due to the elimination of redundancy in search space, the model also had two advantages. First, it essentially changes the least square data-fitting in FHT to the total least squares data-fitting, in which observational errors across all dimensions are taken into accounts, thus more resistant to data noise. Furthermore, it has practically resolved the problem of decreased precision of FHT for target objects mapped to boundaries between accumulators in the parameter space. The second advantage stems from the fact that the integrated parameter space allows an intuitive presentation of vote distribution in the parameter space for 2D and 3D data, which provide an intuitive insight on tuning the parameters and combining multiple instances from the same object.

II. METHOD

Hough transform for line, plane or hyperplane detection in general comprised two steps: input data transformation into parameter space and voting for the accumulators. The workflow of HT is illustrated in Fig. 1.

A. The Introduction of FHT

In 2D data, a point (x, y) can lie on any line $y = mx + c$, $|m| \leq 1$ or $x = m'y + c'$, $|m'| < 1$. The FHT parameter space comprises two sub-spaces (m, c) and (m', c') . Any point (x, y) can be mapped into the first parameter subspace (m, c) into $a_0 + a_1m + a_2c = 0$ ($|m| \leq 1$). The transformation is $a_0 = -y/\sqrt{x^2 + 1}$, $a_1 = x/\sqrt{x^2 + 1}$ and $a_2 = 1/\sqrt{x^2 + 1}$;

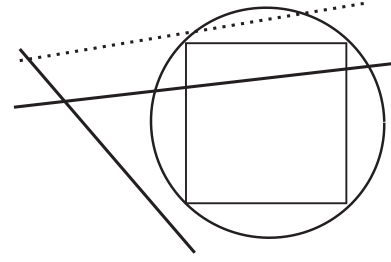


Fig. 2. Tests for the intersection of a line and an accumulator (square) can be relaxed to test the intersection of the line and the circumscribing circle. The approximation only affects the dashed line.

the transformation to map the point into the second parameter subspace (m', c') as line $a_0 + a_1m' + a_2c' = 0$ ($|m'| < 1$) is $a_0 = -x/\sqrt{x^2 + 1}$, $a_1 = y/\sqrt{x^2 + 1}$ and $a_2 = 1/\sqrt{x^2 + 1}$.

FHT then represents each parameter subspace as a nested hierarchy of squares. Each square in the parameter space represents a unique line with certain resolution in the feature space, often called accumulator. Considering an accumulator $m \in [m_1, m_2]$, $c \in [c_1, c_2]$ in the parameter subspace, testing whether a point (x, y) is located on a line $y = mx + c$ where $m \in [m_1, m_2]$, $c \in [c_1, c_2]$ is equivalent to testing whether it is possible that $a_0 + a_1m + a_2c = 0$ where $a_0 = -y/\sqrt{x^2 + 1}$, $a_1 = x/\sqrt{x^2 + 1}$, $a_2 = 1/\sqrt{x^2 + 1}$ and $m \in [m_1, m_2]$, $c \in [c_1, c_2]$. It can be re-formulated as testing the intersection of a square $m \in [m_1, m_2]$, $c \in [c_1, c_2]$ and line $a_0 + a_1m + a_2c = 0$ with known values of a_0, a_1, a_2 . If $[m_1, m_2]$ and $[c_1, c_2]$ have the same length, which can be achieved through data normalization, the test can be relaxed to check whether the line intersects with the circumscribing circle (or hypersphere in high dimension), resulting in dramatically reduction in the computational cost. For example, we only need to test whether $|a_0 + a_1m^* + a_2c^*| < r$, where m^* and c^* are the coordinates of the square's center and r is the radius of the circle (Fig. 2). This approximation could lead to over-counting since a line may intersect the circumscribing circle but not the square. Interestingly, experiments indicate it can significantly reduce the computational complexity in high dimensional data and has trivial effect with the precision of the algorithm.

In general, FHT has the following workflow. Let the input be a set of n -dimensional feature points in data space $\{F_1, F_2, \dots, F_n\}$, with data point j denoted by $F(j) = [F_1(j), F_2(j), \dots, F_n(j)]$. The framework of FHT in a high dimensional data space can be formulated as follows:

- 1) Constructing a k -dimensional parameter space denoted by $\{X_1, X_2, \dots, X_k\}$ with n subspaces and n transformation functions that maps each $F(j)$ into a hyperplane in the parameter space which results in that any hyperplane is represented by

$$a_o(j) + \sum_{i=1}^k a_i(j) X_i = 0, \quad (1)$$

where $a_i(j)$ is a function of $F(j)$ and is normalized such that $\sum_{i=1}^k a_i^2(j) = 1$.

- 2) Defining a threshold for number of vote “ T ” needed to justify a hyperplane and the desired finest resolution “ q ”.
- 3) Dividing the parameter space recursively into hypercubes from low to high resolution, representing with n k -trees, and performing the further analysis only on the hypercubes with votes exceeding a selected threshold.

B. The Integrated Mathematical Model of Hyperplanes by IFHT

IFHT models hyperplanes in the data space in a very intuitive way. Let the input be a set of n -dimensional feature points in data space $\{F_1, F_2, \dots, F_n\}$, with data point j denoted by $F(j) = [F_1(j), F_2(j), \dots, F_n(j)]$. A hyperplane can be modelled as

$$\sum_{i=1}^n \beta_i F_i + \beta_{n+1} = 0$$

$$\text{where } \sum_{i=1}^n (\tau_i \cdot \beta_i)^2 = 1 \text{ and } \beta_1 \geq 0 \quad (2)$$

Here τ_i is a scale factor for normalization purpose, and the way to select it will be discussed later. For simplicity, we here relax the restriction of the model by dropping the requirement $\beta_1 \geq 0$. Suppose we know the initial ranges of $\{\beta_1, \beta_2, \dots, \beta_{n+1}\}$. Assume the half-length of the ranges are $\{L_1, L_2, \dots, L_{n+1}\}$. We have

$$\sum_{i=1}^n \frac{F_i(j)}{W(j)} \frac{L_i}{L_{n+1}} \frac{\beta_i}{L_i} + \frac{\beta_{n+1}}{W(j) L_{n+1}} = 0 \text{ for all } j \in \Omega$$

Let $X_i = \frac{\beta_i}{L_i}$. The transformation for a data point $F(j) = [F_1(j), F_2(j), \dots, F_n(j)]$ to parameter space $\{X_1, X_2, \dots, X_{n+1}\}$ is

$$\sum_{i=1}^{n+1} a_i(j) X_i = 0$$

$$\text{where } a_i(j) = \frac{F_i(j)}{W(j)} \frac{L_i}{L_{n+1}}, \text{ and } a_{n+1}(j) = \frac{1}{W(j)}$$

Here $W(j)$ is a weighted scale used to ensure $\sum_{i=1}^{n+1} a_i^2(j) = 1$. In this new parameter space, each dimension X_i is in $[-1, 1]$ and the parameter space can be quantized into hypercubes for voting. In addition, the requirement of $\beta_1 \geq 0$ can be easily satisfied by dropping specific hypercubes with $X_1 < 0$ as $X_1 = \frac{\beta_1}{L_1}$.

Eq. (2) is the full model of IFHT, which allows targeted searching, i.e., searching objects within a specific part of the space. For the general application aiming for all objects in the space, (2) can be simplified into

$$\sum_{i=1}^n \beta_i F_i + \beta_{n+1} = 0$$

$$\text{where } \sum_{i=1}^n \beta_i^2 = 1, \beta_1 \geq 0$$

Here β_{n+1} is limited and can be quantified into small intervals for investigation. It can be easily proved that the β_{n+1} is smaller

than or equal to the maximum distance D of data points $\{F_1, F_2, \dots, F_n\}$ to the origin in the space coordinate system.

For simplicity, we can rewrite the above equation as

$$\sum_{i=1}^n \beta_i F_i + d \cdot \beta_{n+1} = 0$$

$$\text{where } \sum_{i=1}^n \beta_i^2 = 1, \beta_1 \geq 0 \text{ and } |\beta_{n+1}| \leq 1 \quad (3)$$

where d equals to the maximum distance D of data points $\{F_1, F_2, \dots, F_n\}$ to the origin in the space coordinate system. In practice, d could be a bit smaller, for example $0.8D$, for the better resolution.

For example, in a 2D data space, (3) is $\beta_1 x + \beta_2 y + d \cdot \beta_3 = 0$, $\beta_1^2 + \beta_2^2 = 1$, $|\beta_i| \leq 1, i = 1, 2, 3$. We then use $(\beta_1, \beta_2, \beta_3)$ as the IFHT parameter space. The transformation to map a point (x, y) into the parameter space $a_1 \beta_1 + a_2 \beta_2 + a_3 \beta_3 = 0$ with $\beta_1^2 + \beta_2^2 = 1, |\beta_3| \leq 1$ is

$$a_1 = x / \sqrt{x^2 + y^2 + d^2}$$

$$a_2 = y / \sqrt{x^2 + y^2 + d^2}$$

$$a_3 = d / \sqrt{x^2 + y^2 + d^2}$$

Here $|\beta_i| \leq 1$ for $i = 1, 2, 3$, so the parameter space $(\beta_1, \beta_2, \beta_3)$ has the same length along each dimension and can be quantized into cubes for voting.

In the above explanation, the dimension of the parameter space is $n+1$ for the n -dimensional input data. This is not necessarily so. For example, if we are interested in target objects (lines, planes or hyperplanes) with zero intercept, i.e., objects that pass through the origin, the IFHT parameter space will be n -dimensional. We hereafter denote k as the dimension of the IFHT parameter space for generality.

C. Hierarchical Vote Counting Scheme

Every point in the data space corresponds to one hyperplane in the IFHT parameter space. To achieve a robust detection, small perturbation in the parameter should be allowed to deal with the noise inside the data. This can be easily implemented in IFHT by evaluating a small interval along each dimension in the parameter space rather than a specific value. The length of interval then corresponds to the required resolution of the target hyperplane (straight line for 2D space). To test whether a hyperplane consisted of more than T points exists in the data space, IFHT quantified the parameter space into small accumulators and let each data point to vote for them. Since the IFHT parameter space has the same length along each dimension, the parameter space can be quantized into small accumulators (squares, cubes, or hypercubes) with the same length along each dimension too.

IFHT parameter space is linear. If an accumulator collects T votes, no any sub-accumulator can surpass T votes. This allows the IFHT parameter space to be quantized hierarchically. We can divide IFHT parameter space into several large accumulators for voting and then divide only those with enough votes into smaller

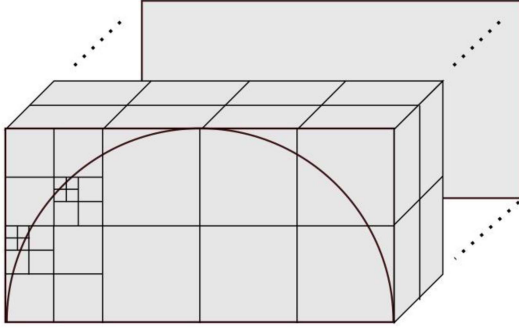


Fig. 3. The coarse-to-fine strategy for quantized accumulators of IFHT. Accumulators have the same length along each dimension.

accumulators for further testing until the required resolution is achieved. This coarse-to-fine strategy pose great advantage compared to other polar-coordinate based Hough transform methods.

As the accumulator has the same length along each dimension, we can use the same approximation method shown in Fig. 2 for vote testing. The test equation is

$$\left| \sum_{i=1}^k a_i(j) C_i \right| \leq r \quad (4)$$

where $[C_1, C_2, \dots, C_k]$ is the center of accumulator and r is the radius of the hypersphere circumscribing the accumulator (hypercube).

D. A Single K -Tree Representation

In IFHT, the parameter space is a hollow hyper-semi-cylinder (Fig. 3). We represent the parameter space with a nested hierarchy hypercube and associate a k -tree with it. The root node of the tree corresponds to a hypercube centered at a vector C_0 with side length S_0 . Each node of the tree has 2^k children arising when that node's hypercube is halved along each of its k dimensions. A vector $\mathbf{b} = [b_1, \dots, b_k]$ where each b_i is -1 or 1 is used as an index for a child of a node. If a node at level l has center C_l , then the center of its child with index $[b_1, \dots, b_k]$ is

$$C_l + \frac{S_{l+1}}{2} \cdot \mathbf{b}$$

where S_{l+1} is the length of child at level $l+1$ and $S_{l+1} = S_l/2$.

IFHT borrows the coarse-to-fine mechanism from FHT. In detail, for a hypercube at level l centered at $[C_{l1}, C_{l2}, \dots, C_{lk}]$, the distance D_l to a hyperplane defined in (3) can be estimated as $D_l = \sum_{i=1}^k a_i(j) C_{li}$. By normalizing distance with side-length S_l , we have

$$R_l = \sum_{i=1}^k a_i(j) C_{li}/S_l \quad (5)$$

We can iteratively calculate this normalized distance. Suppose we know the initial ranges of $\{X_1, X_2, \dots, X_k\}$ in (3) are centered

at $C_0 = [C_{01}, C_{02}, \dots, C_{0k}]$, we have

$$R_0 = \sum_{i=1}^k a_i(j) C_{0i}/S_0$$

$$R_{l+1} = 2R_l + \frac{1}{2} \sum_{i=1}^k a_i(j) b_i \quad (6)$$

The test criterion in 4) can be simplified as

$$|R| \leq \sqrt{k}/2 \quad (7)$$

In (2), we use constraint $\sum_{i=1}^{k-1} (\tau_i \cdot \beta_i)^2 = 1$, $k = n+1$ to guarantee that the hyperplane in the data space is unique. When neither prior information nor preference about where the target hyperplane exists, the initial ranges of $\{\beta_1, \dots, \beta_{k-1}\}$ are $|\beta_i| \leq 1$, then the constraint of $\sum_{i=1}^{k-1} (\tau_i \cdot \beta_i)^2 = 1$ can be simplified into

$$\sum_{i=1}^{k-1} \beta_i^2 = 1 \quad (8)$$

In the parameter space, this is equivalent to test whether an accumulator intersect with $(k-1)$ -dimensional unit hypersphere. Following the same principle of the simplification used for (4), for an accumulator centered at $[C_1, C_2, \dots, C_k]$ at level l , with r the radius of the hypersphere circumscribing the accumulator (hypercube), the test equation is

$$\left| \sum_{i=1}^{k-1} C_i^2/S_0 - 1 \right| \leq \sqrt{k}/2^l \quad (9)$$

E. Workflow for IFHT

In data space $\{F_1, F_2, \dots, F_n\}$, when giving a set of data point $F(j) = [F_1(j), F_2(j), \dots, F_n(j)]$, IFHT parameters that need to be predetermined are as follows:

- 1) The minimum votes count “ T ” as threshold and the desired finest resolution “ q ”.
- 2) A transformation that maps each $F(j)$ into the parameter space $\{X_1, X_2, \dots, X_k\}$ which results in that any hyperplane is represented by

$$\sum_{i=1}^k a_i(j) X_i = 0 \quad (10)$$

where $a_i(j)$ is a function of $F(j)$ and is normalized so that $\sum_{i=1}^k a_i^2(j) = 1$. For a typical case where no preference or prior information exists, the target hyperplanes can be modelled as $\sum_{i=1}^n \beta_i F_i + d \cdot \beta_{n+1} = 0$, $\sum_{i=1}^n \beta_i^2 = 1$, $\beta_1 \geq 0$, where d is a scale smaller than or equal to the maximum distance of data points to the origin. The mapping function for (10) is $k = n + 1$, $a_i(j) = \frac{F_i(j)}{W(j)^d}$, $j = 1, \dots, n$, and $a_{n+1}(j) = \frac{1}{W(j)}$ where $W(j)$ is a weighted scale used to insure $\sum_{i=1}^{n+1} a_i^2(j) = 1$.

IFHT can be summarized as follows:

- 1) compute the initial normalized distance from all hyperplane to the root node and count its votes. The root node is

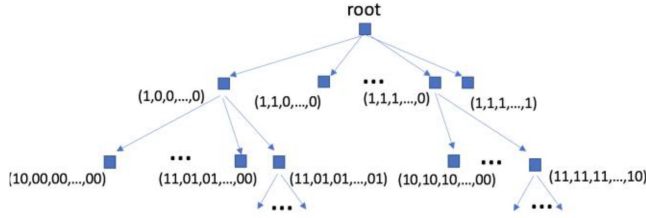


Fig. 4. The k -tree representation of the parameter space. For simplicity, we here use 0 and 1 for binary values of vector \mathbf{b} . Each node can be subdivided into 2^k child nodes except that there are only 2^{k-1} child nodes for the root node as $b_1 = 1$ due to $\beta_1 \geq 0$.

usually centered at $(0, \dots, 0)$ with half of the side-length as $(1, \dots, 1)$.

- 2) for each child node, calculate whether it intersects with k -dimensional unit hypersphere with (9). If not, stop processing that child node. If yes, calculate the normalized distance for that child node with (6) and vote for that child node with (7). If it has vote count larger than T , subdivide it into k -tree child nodes. For the child nodes of the root node, only those with first index $b_1 = 1$ are kept for further analysis.
- 3) When there is no node with the resolution equal to q and the vote count larger than T , record the node with the highest resolution. Otherwise, output all nodes with the resolution equal to q and the vote count larger than T .

F. The Complexity of IFHT

In FHT, it has been proved that the number of active accumulators does not increase exponentially with the quantization level and the dimensionality of the parameter space. This is fully guaranteed by the “thin tree” theorem. In IFHT, the “thin-tree” theorem is as follows:

Thin tree Theorem: For a hyperplane with M data points, each data point is transformed with (2) into a hyperplane in the parameter space and M hyperplanes in the parameter space intersect at a single point P . Assume they are uniformly distributed in orientation at the point P . Given a minimum vote threshold T , the number of active accumulators of size q that receives T or more votes is less than a constant number K that does not depend on q .

Proof: For IFHT, if we drop the constraint $\sum_{i=1}^n (\tau_i \cdot \beta_i)^2 = 1$ and $\beta_1 \geq 0$, the transform can be regarded as a special instance of FHT in $(n+1)$ -dimension. If M hyperplanes (corresponding to the M data points in the data space) in the parameter space intersect at point P with a uniformly distribution in orientation, it satisfies FHT “thin tree” theorem. As the constraint only decreases the number of active accumulators to be tested, so the number of active accumulators of size q that receive T or more votes is less than a constant number K that does not depend on q .

In Fig. 5, we collected the overall number of active accumulators by IFHT for two 3D datasets. It is clear that in both examples, the number of active accumulators does not increase exponentially with the quantization level.

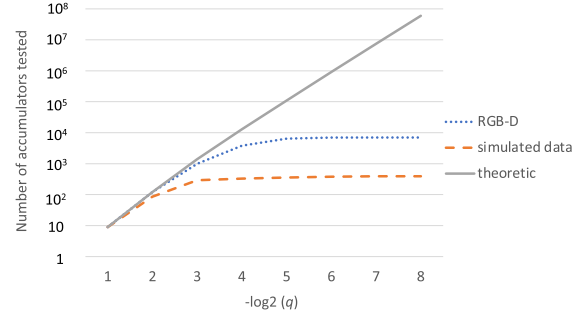


Fig. 5. The number of active accumulators by IFHT in two 3D datasets and the theoretic number of accumulators.

G. Two Possible Variants of IFHT

In IFHT described above, a hyperplane is modelled as

$$\sum_{i=1}^n \beta_i F_i + \beta_{n+1} = 0$$

$$\text{where } \sum_{i=1}^n (\tau_i \cdot \beta_i)^2 = 1 \text{ and } \beta_1 \geq 0$$

It is possible to modify the equation slightly in certain scenarios. One modification could be to put the intercept β_{n+1} of the line into the normalization part. This would relieve us from specifying the range parameter for β_{n+1} which often requires calculating the maximum distance of data points to the origin. The corresponding mathematical model is as follows:

$$\sum_{i=1}^n \beta_i F_i + \beta_{n+1} = 0$$

$$\text{where } \sum_{i=1}^{n+1} (\tau_i \cdot \beta_i)^2 = 1 \text{ and } \beta_1 \geq 0$$

In addition, the implementation of IFHT algorithm could be modified slightly. In the IFHT mathematic model, $n+1$ variables have been used. However, when $\{\beta_2, \dots, \beta_{n+1}\}$ is determined, β_1 can be uniquely decided and serves as a dummy variable. It is possible to calculate β_1 straightforwardly without putting it into the parameter space in the IFHT implementation.

III. RESULTS

In all variants of Hough transform, to determine the desired the quantization level q is not a trivial task. When a very coarse quantization level with a small value of q is chosen, too many lines, planes or hyperplanes could be reported which cannot satisfy the required precision. When a large value of q is chosen, the final results could be very sensitive to statistical noise or quantization noise in the data. In FHT, it is very frequently observed the same lines, planes or hyperplanes are reported as multiple instances by several accumulators at a specific level l , but no lines or hyperplanes can be found at level $l+1$. As a consequence, to automatically report how many lines or hyperplanes inside the high-dimensional data with HT or FHT is still challenging.

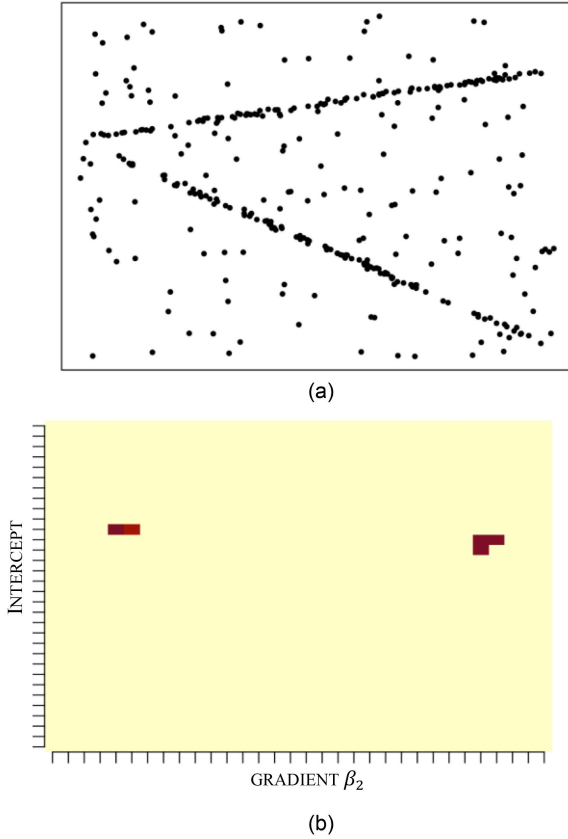


Fig. 6. IFHT provides an intuitive insight for determining the number of lines in an image and parameter tuning. (a) An image with two lines. (b) IFHT vote distribution in the parameter space shows two clusters, corresponding two lines in the data. Here the IFHT parameter space in quantitation level $l = 5$ is shown.

In IFHT, the integrated parameter space allows an intuitive presentation of vote distribution in the parameter space, which provides an intuitive insight on combining detected lines or tuning the parameters as shown in Fig. 7 for 2D data and Fig. 8 for 3D data. It is very clear that when a coarse resolution is chosen, the multiple instances from the same line or hyperplane always cluster together. This observation is a manifest of IFHT thin-tree theorem in the parameter space. It provides a very robust and intuitive way to determine the number of objects to be reported, and it is also instructive in parameter tuning and merging of multiple instances if necessary.

The proposed IFHT method has some interesting properties. In FHT, the noise in different dimension is not treated equally. For example, in 2D data, $y = mx + c$, $|m| \leq 1$ and $x = m'y + c'$, $|m'| < 1$ are used for modeling hyperplanes. When fitting the model to available data points, FHT minimizing the error parallel to the y axis, as shown in Fig. 8(a). IFHT treats each dimension of data equally, and essentially changes the least square data-fitting in FHT to the total least-square data-fitting [22], in which observational errors in both dependent and independent variables were taken into account. In IFHT, we test whether it is possible that $\beta_1 x + \beta_2 y + \beta_3 = 0$ where $\beta_1^2 + \beta_2^2 = 1$ for a specific pair of β_1, β_2 . This is achieved through calculating the direct distance a square to a line in 2D case. The data-fitting can be illustrated in Fig. 8(b).

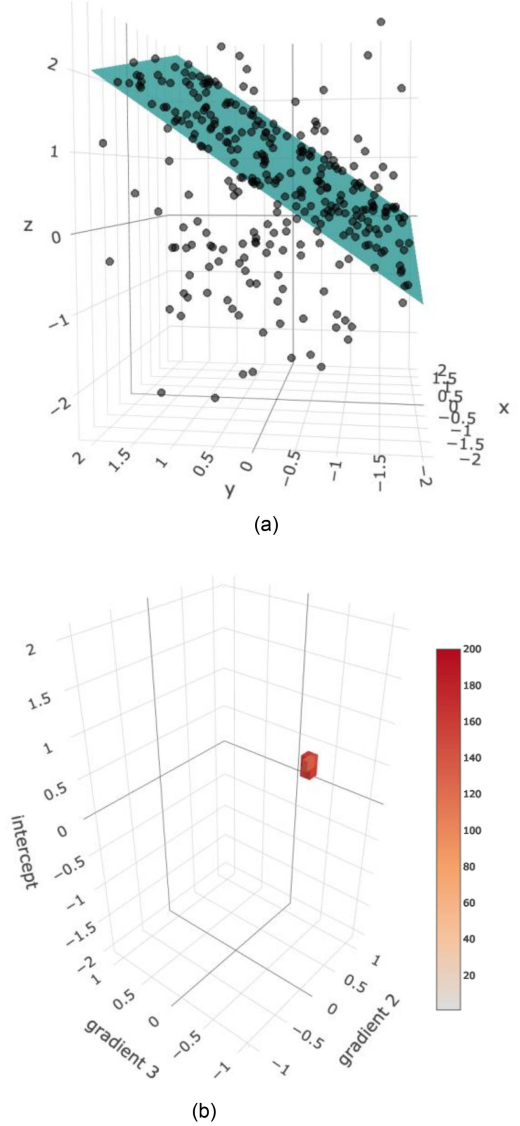


Fig. 7. A 3D data space and its IFHT parameter space. (a) 3D data points and the existing plane. (b) The IFHT vote distribution in the parameter space shows a cluster, corresponding the plane in the data space. Here the IFHT parameter space in quantitation level $l = 5$ is shown.

We compared the performances of our IFHT method and HFT with simulated data using Monte Carlo method. In the first experiment, a specific number N of data points from a line in 2D space were generated, with $x \sim U(-2, 2)$, $y = 0.8x + 0.24 + g$. Here g is Gaussian white noise with standard deviation 0.04 and 0.08. Then the same number of random data points with $x, y \sim N(0, 1)$, without those in the proximity of the line, were generated for the background data points. For each set of parameters, 100 images were generated. IHFT and HFT with the same parameters ($T = 0.75N$ and $l = 5$) were used to detect the line in each image. The number of correctly reported points on the detected line were recorded and the ratio to the overall number of true points of the line is reported as detection rate. Fig. 9(a) shows that in both scenarios, the performance of IHFT is significantly better than that of FHT.

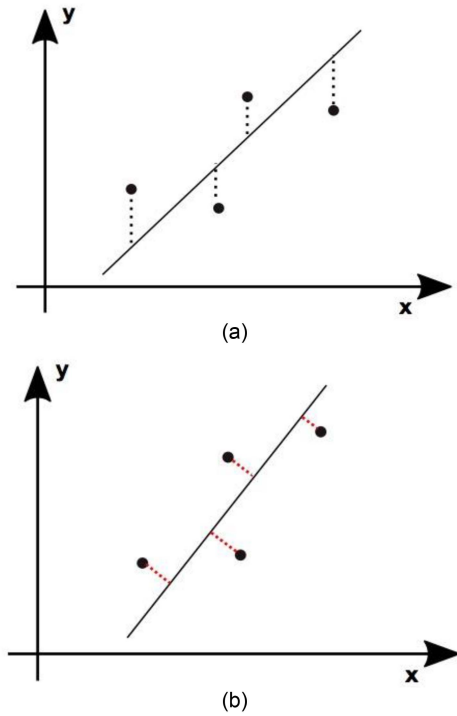


Fig. 8. IFHT and FHT model different error distances. (a) FHT minimizes the sum of L1 distances (dashed line in black) of points involved to the target line. (b) IFHT minimizes the sum of Euclidian distances (dashed line in red) of points involved to the target line.

It has been noted that when the initial parameters have been chosen, the positioning and size of accumulators in FHT will be fixed. In some cases, peaks may straddle the boundaries between accumulators [21]. Due to the different parameter spaces have been used, only very few lines cause this type of problem to both FHT and IFHT methods. Line $y = 0.5$ is one of them when parameters ($l = 5$ and $d = 2$) are used in both methods. Fig. 9(b) shows the detection rate of both methods when the standard deviation of the Gaussian white noise is 0.04 and 0.08, respectively. The performance of IFHT was not significantly affected, while the performance of FHT was indeed seriously undermined in this scenario.

In above simulations, the white noise has only been added into the second dimension of the observed data. This would be rare in the real data since the noise usually distributed across both dimensions of observed data. We repeated the first experiment with noise presented across both dimensions. The observed data point for the line is $t \sim U(-2, 2)$, $x = t + g1$, $y = 0.8t + 0.24 + g2$. Here $g1$ and $g2$ are Gaussian white noise with the same standard deviation. We tested IFHT and FHT using with the same parameters ($T = N/2$ and $l = 5$). Their performances are shown in Fig. 10. Apparently, IFHT has significantly better performances than FHT.

We further tested the performance of IFHT for the real 3D data. An RGB-D image from a publicly available image database [23] was chosen. The RGB image and the corresponding depth image are shown in Fig. 11(a) and (b) respectively. The edge

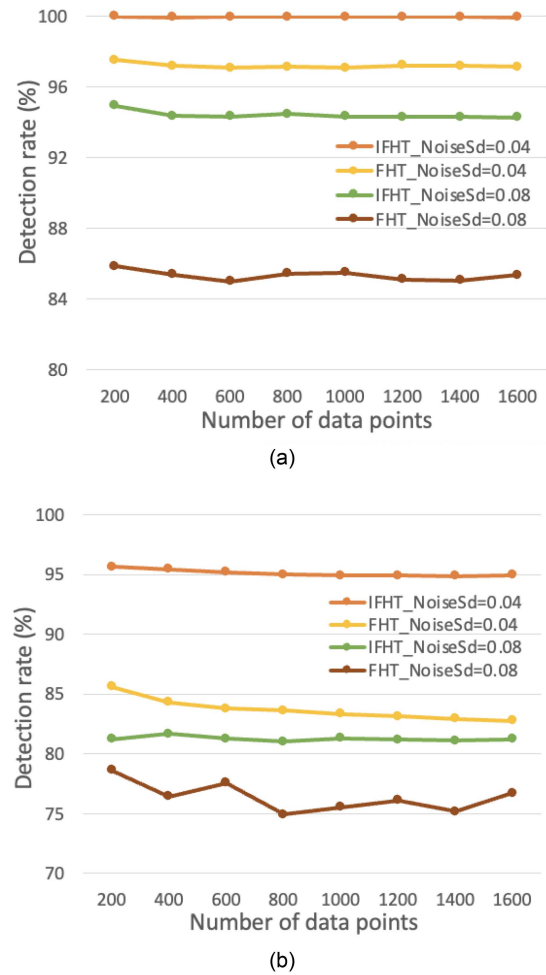


Fig. 9. The comparison of performances between IFHT and FHT with two levels of measurement noise in the second dimension. (a) A typical line detection experiment; (b) a challenging scenario where peaks in the parameter space straddle the boundaries between accumulators in both IFHT and FHT methods.

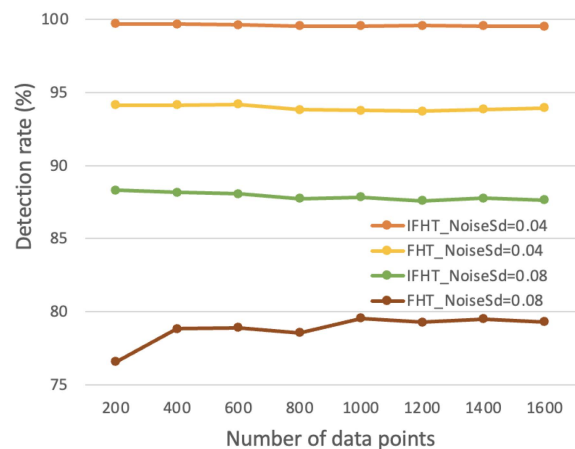


Fig. 10. The comparison of detection rates between IFHT and FHT with measurement noise in both dimensions of the observed data. Two levels of noise, with standard deviation 0.02 and 0.067 respectively, have been tested.

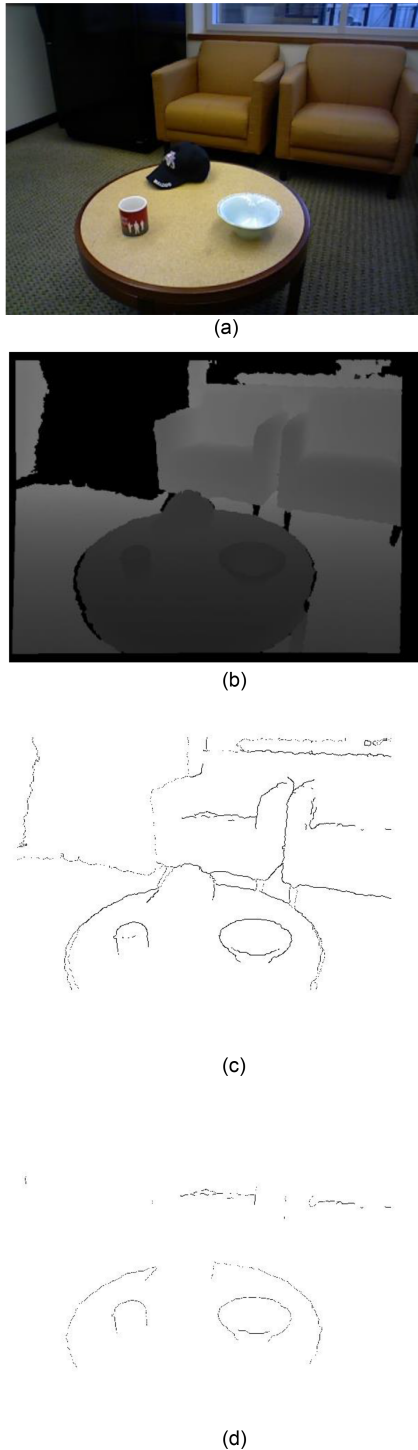


Fig. 11. The performance of IFHT on an RGB-D image. (a) The RGB image, (b) the depth image, (c) The edge of objects detected using the Canny operator with the depth image, (d) The detected plane in 3D space by IFHT.

of objects was detected using the Canny operator implemented in OpenCV Fig. 11(c), and then transformed into 3D data points using the transformation suggested by Kinect. The input was centered and scaled with standard deviation along each dimension (by R function “scale”). IFHT was then applied with parameters ($T = 600$, $l = 7$ and the range of intercept = 4). IFHT

successfully detected the plane containing the edge of the table and the inner surface of the sofa as shown in Fig. 11(d).

IV. CONCLUSION

The detection of line, plane or hyperplane in multidimensional data has many applications in computer vision, data clustering and artificial intelligence. Though there are many 2D algorithms, few of them are practical for high-dimensional data due to the increased computational and storage demands. The Fast Hough transform approach suggested by Li et al. is so far one of the most efficient with relative space efficiency $R_s = (\alpha/\beta)^k$ for a k dimensional case. However, the mathematical model used by FHT imposes a restriction that multiple independent parameter subspaces need to be constructed and searched. The burden of dealing with multiple parameter subspaces not only increases the complexity of algorithm design and implementation, but also makes the intuitive visualization of the parameter space almost impossible. As a consequence, the parameter tuning of FHT is difficult and there is lack of an efficient method to merge multiple instances from the same object (line, plane or hyperplane). In addition, the sensitivity and precision of FHT vary with the parameters of target objects. Lines, planes and hyperplanes which are mapped to the boundaries of accumulators in the parameter space could have very low precision. These disadvantages have seriously limited the application of FHT algorithm.

We have proposed an integrated mathematical model for Hough transform, corresponding to an integrated fast Hough transform parameter space. The parameter space of IFHT can be represented with a single k -tree to support hierarchical searching. Apart from the obvious advantage on simplicity of algorithm design and reduces redundancy in computation and storage, the model allows the intuitive presentation of the parameter space. This new insight can be used to decide the number of lines, planes or hyperplanes in the data, and to help with tuning the parameters and combining detected instances if necessary. Furthermore, IFHT essentially changes the least square data-fitting in FHT to the total least squares data-fitting, in which observational errors across all dimensions are taken into accounts, thus more resistant to data noise. In our test, IFHT successfully avoided the drawback of varied sensitivity and precision embed in FHT algorithm, with very high precision for almost all lines or planes in the data space. In almost all simulated images with different levels of noise and parameters, IFHT surpassed FHT in terms of robustness and precision significantly. IFHT also shows good performance for the real 3D data.

ACKNOWLEDGMENT

The author would like to thank J. Xu for his technical support.

REFERENCES

- [1] P. V. C. Hough, “Method and means for recognizing complex patterns,” USA Patent US3069654A, 1962.
- [2] S. Kundu, S. Malakar, Z. W. Geem, Y. Y. Moon, P. K. Singh, and R. Sarkar, “Hough transform-based angular features for learning-free handwritten keyword spotting,” *Sensors*, vol. 21, no. 14, Jul. 7, 2021, Art. no. 4648, doi: [10.3390/s21144648](https://doi.org/10.3390/s21144648).

- [3] M. Kowdiki and A. Khaparde, "Adaptive hough transform with optimized deep learning followed by dynamic time warping for hand gesture recognition," *Multimedia Tools Appl.*, vol. 81, pp. 2095–2126, Oct. 20, 2021, doi: [10.1007/s11042-021-11469-9](https://doi.org/10.1007/s11042-021-11469-9).
- [4] X. H. Cao, D. F. Liu, and X. Y. Ren, "Detection method for auto guide vehicle's walking deviation based on image thinning and Hough transform," *Meas. Control*, vol. 52, no. 3-4, pp. 252–261, Mar./Apr. 2019, doi: [10.1177/0020294019833073](https://doi.org/10.1177/0020294019833073).
- [5] K. Zhao, Q. Han, C. B. Zhang, J. Xu, and M. M. Cheng, "Deep Hough transform for semantic line detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4793–4806, May 3, 2021, doi: [10.1109/TPAMI.2021.3077129](https://doi.org/10.1109/TPAMI.2021.3077129).
- [6] X. Gan, A. W. Liew, and H. Yan, "Discovering biclusters in gene expression data based on high-dimensional linear geometries," *BMC Bioinf.*, vol. 9, Apr. 23, 2008, Art. no. 209, doi: [10.1186/1471-2105-9-209](https://doi.org/10.1186/1471-2105-9-209).
- [7] E. Achtert, C. Böhm, J. David, P. Kröger, and A. Zimek, "Global correlation clustering based on the Hough transform," *Stat. Anal. Data Mining: ASA Data Sci. J.*, vol. 1, no. 3, pp. 111–127, 2008, doi: [10.1002/sam.10012](https://doi.org/10.1002/sam.10012).
- [8] H. Kalviainen, P. Hirvonen, L. Xu, and E. Oja, "Probabilistic and nonprobabilistic hough transforms - overview and comparisons," *Image Vis. Comput.*, vol. 13, no. 4, pp. 239–252, May 1995, doi: [10.1016/0262-8856\(95\)99713-B](https://doi.org/10.1016/0262-8856(95)99713-B).
- [9] R. Yuan, J. Zhang, and J. M. Zuo, "Lattice strain mapping using circular Hough transform for electron diffraction disk detection," *Ultramicroscopy*, vol. 207, Dec. 2019, Art. no. 112837, doi: [10.1016/j.ultramic.2019.112837](https://doi.org/10.1016/j.ultramic.2019.112837).
- [10] Q. K. Liang et al., "Angle aided circle detection based on randomized Hough transform and its application in welding spots detection," *Math. Biosciences Eng.*, vol. 16, no. 3, pp. 1244–1257, 2019, doi: [10.3934/mbe.2019060](https://doi.org/10.3934/mbe.2019060).
- [11] Z. Zalevsky, G. S. Buller, T. Chen, M. Cohen, and R. Barton-Grimley, "Light detection and ranging (LiDAR): Introduction," *J. Opt. Soc. Amer. A: Opt. Image Sci. Vis.*, vol. 38, no. 11, pp. L1D1–L1D2, Nov. 1, 2021, doi: [10.1364/JOSAA.445792](https://doi.org/10.1364/JOSAA.445792).
- [12] I. Vasile, E. Tudor, I. C. Sburlan, M. A. Gheti, and G. Popa, "Experimental validation of LiDAR sensors used in vehicular applications by using a mobile platform for distance and speed measurements," *Sensors*, vol. 21, no. 23, Dec. 6, 2021, Art. no. 8147, doi: [10.3390/s21238147](https://doi.org/10.3390/s21238147).
- [13] Y. F. Tian, W. Song, L. Chen, Y. Sung, J. Kwak, and S. Sun, "Fast planar detection system using a GPU-based 3D hough transform for LiDAR point clouds," *Appl. Sci.-Basel*, vol. 10, no. 5, Mar. 2020, Art. no. 1744, doi: [ARTN 1744 10.3390/app10051744](https://doi.org/10.3390/app10051744).
- [14] W. Song, L. F. Zhang, Y. F. Tian, S. Fong, and S. Sun, "3D hough transform algorithm for ground surface extraction from LiDAR point clouds," in *Proc. IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber Phys. Social Comput. IEEE Smart Data*, 2019, pp. 916–921, doi: [10.1109/iThings/GreenCom/CPSCoM/SmartData.2019.00163](https://doi.org/10.1109/iThings/GreenCom/CPSCoM/SmartData.2019.00163).
- [15] F. A. Limberger and M. M. Oliveira, "Real-time detection of planar regions in unorganized point clouds," *Pattern Recognit.*, vol. 48, no. 6, pp. 2043–2053, Jun. 2015, doi: [10.1016/j.patcog.2014.12.020](https://doi.org/10.1016/j.patcog.2014.12.020).
- [16] H. W. Li, M. A. Lavin, and R. J. Lemaster, "Fast Hough transform - a hierarchical approach," *Comput. Vis. Graph.*, vol. 36, no. 2-3, pp. 139–161, Nov./Dec. 1986, doi: [10.1016/0734-189x\(86\)90073-3](https://doi.org/10.1016/0734-189x(86)90073-3).
- [17] S. M. Karpenko and E. I. Ershov, "Analysis of properties of dyadic patterns for the fast Hough transform," *Problems Inf. Transmiss.*, vol. 57, no. 3, pp. 292–300, Jul. 2021, doi: [10.1134/S0032946021030078](https://doi.org/10.1134/S0032946021030078).
- [18] M. L. Brady, "A fast discrete approximation algorithm for the Radon transform," *SIAM J. Comput.*, vol. 27, no. 1, pp. 107–119, Feb. 1998, doi: [10.1137/S0097539793256673](https://doi.org/10.1137/S0097539793256673).
- [19] X. Gan, A. W.-C. Liew, and H. Yan, "Representation and extraction of biclusters from data arrays," USA Patent US7849088B2, 2008.
- [20] "Gandalf: The fast computer vision and numerical library," 2006. [Online]. Available: <http://gandalf-library.sourceforge.net>
- [21] J. Illingworth and J. Kittler, "The adaptive Hough transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 690–698, May 1987, doi: [10.1109/tpami.1987.4767964](https://doi.org/10.1109/tpami.1987.4767964).
- [22] I. Markovsky and S. Van Huffel, "Overview of total least-squares methods," *Signal Process.*, vol. 87, no. 10, pp. 2283–2302, Oct. 2007, doi: [10.1016/j.sigpro.2007.04.004](https://doi.org/10.1016/j.sigpro.2007.04.004).
- [23] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 3050–3057.



Yanhui Li is currently working toward the PhD degree, supervised by Xiangchao Gan with the Nanjing Agricultural University. Her research interests include pattern recognition, high dimensional data analysis, and deep learning for breeding research.



Xiangchao Gan (Member, IEEE) received the BA degree from the Nanjing University of Posts and Telecommunications, the MEng degree from Xi'an Jiaotong University, and the PhD degree from the City University of Hong Kong. He was a postdoc with the Department of Computer Science, King's College London from 2007 to 2009. He served as a statistical geneticist with the Wellcome Trust Center for Human genetics, Oxford University from 2009 to 2013. He was a group leader in Max-Planck Institute for Plant Breeding Research, Cologne, Germany from 2013 to

2020. He is currently a professor with Nanjing Agricultural University, China. His research interests include pattern recognition, machine learning, statistical genetics, and bioinformatics.