

Received 24 May 2023, accepted 13 July 2023, date of publication 19 July 2023, date of current version 25 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296800

RESEARCH ARTICLE

Exploiting Sequence Analysis for Accurate Light-Field Depth Estimation

LEI HAN^{1,2}, SHENGNAN ZHENG^{1,2}, ZHAN SHI¹, AND MINGLIANG XIA¹

¹School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China

²Jiangsu Province Engineering Research Center of IntelliSense Technology and System, Nanjing Institute of Technology, Nanjing 211167, China

Corresponding author: Lei Han (hanl@njit.edu.cn)


This work was supported in part by the National Natural Science Foundation of China under Grant 62076122, in part by the Natural Science Foundation of the Nanjing Institute of Technology under Grant ZKJ201906, and in part by the Open Foundation from the Jiangsu Province Engineering Research Center of IntelliSense Technology and System under Grant ITS202101 and Grant ITS202201.

ABSTRACT Depth estimation for light field (LF) images is the cornerstone of many applications of light field cameras, such as 3D reconstruction, defects inspection, face liveness detection, and so forth. In recent years, convolutional neural network (CNN) has dominated the primary workhorse for depth estimation. However, the interpretability of the network and the accuracy of the depth estimation results still need to be improved. This paper uses the conditional random field (CRF) theory to explain and model the LF depth estimation. Further, from the perspective of sequence analysis, we extract the sequence features of epipolar plane image (EPI) patches with recurrent neural network (RNN) and serve as the unary term of the energy function in the CRF. Then, a unified neural network (called as LFRNN) is designed to solve the CRF and get the disparity map. Our LFRNN builds upon two-stage architecture, involving a local depth estimation and a depth refinement. In the first part, we design an RNN to analyze the vector sequences in EPI patches and obtain local disparity values. There are two thinking behind the design of this part. The first is the general principle that the slope of the straight line in the EPI is inversely proportional to the depth; the second is our unique observation that those straight lines are distributed in vector sequences. In the second part, continuous CRF is used to optimize the output of the first part. We train LFRNN on a synthetic LF dataset and test it on both synthetic and real-world LF datasets. Quantitative and qualitative results validate the superior performance of our LFRNN over the state-of-the-art methods.

INDEX TERMS Computer vision, depth estimation, light field imaging, deep learning, sequence analysis.

I. INTRODUCTION

Depth information is vital in computer vision applications such as 3D reconstruction, robot vision, and semantic segmentation [1]. With the availability of light field (LF) cameras in the consumer market, estimating depth from LF has attracted significant interest from both academia and industry [2], [3]. LF camera captures rich spatial-angular data of 3D scenes by its distinct imaging structure, placing a microlens array after the main lens. Those data can be reconstructed into refocusing images, sub-aperture images (SAIs), and epipolar plane images (EPIs). EPIs render regular line patterns with different slopes, and there is a geometry

The associate editor coordinating the review of this manuscript and approving it for publication was K. C. Santosh .

relationship between the straight-line slope and the depth value [4].

Researchers have proposed many LF depth estimation methods using the above property of EPIs. Classic methods estimate depth with manual features. Structural tensor [5], [6], [7], [8], parallelogram [9], [10], sparse representation [11], [12], and other operators are designed to measure the line slopes and further compute depth value with optimization technology. Recently, LF depth estimation has made remarkable achievements that benefited from the rising of deep learning. In order to use the multi-view information in LF, some learning-based methods [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] integrate multi-directional EPIs by multi-stream network architecture [17], [18], [21], [22]. Furthermore, the visual attention mechanism is also

introduced into LF depth estimation to improve the sharpness of the edge [20], [21]. Through these means to improve the structure of the neural network, the learning-based methods are superior to the classic methods in the accuracy and robustness of depth estimation. However, the network model also has shortcomings, such as weak interpretability, difficult generalization, and occlusion processing. Significantly, the existing methods pay more attention to the texture of LF images while ignoring LF's sequence characteristics. Hence, convolutional neural networks (CNNs) are generally utilized as the backbone. Such CNN frameworks may become a fetter for scholars to improve the accuracy of depth estimation, especially in occlusion processing.

This paper proposes a novel two-stage light field depth estimation network from the perspective of sequence analysis. The first stage is to get a local depth map by a recurrent neural network (RNN). The local depth map is subsequently refined in a conditional random field (CRF) framework. The main contributions of this paper are two-fold:

(1) We utilize CRF theory to model the problem of LF depth estimation and design a unified network framework by deep learning. The theory of CRF is highly interpretable and has long been a cornerstone in traditional computer vision methods. In recent years, there has been a growing trend towards integrating CRFs with deep learning techniques, leading to exciting new developments. In the context of light field image processing, this paper aims to construct a depth estimation model by leveraging the power of CRFs and proposes a comprehensive solution framework based on deep learning. By combining the interpretability of CRFs and the representation learning capabilities of deep learning, this work strives to advance the state-of-the-art in depth estimation for LF images.

(2) We notice a notable pattern that the straight lines within EPI are arranged in the vector sequence. Consequently, we put forth the concept of local estimation through sequence analysis, employing RNN for implementation. While it has been proven that the slope of the straight lines in the EPI is inversely proportional to the corresponding depth value, our method takes a fresh approach by learning this relationship through sequence analysis.

The remainder of this paper is organized as follows. After reviewing related works in Section II, we detail in Section III the geometry principle of LF imaging and indicate the sequence characterization in the EPI patch. In Section IV, we propose a novel depth estimation method based on the CRF model and the idea of sequence analysis. To compare with the state-of-the-art methods of depth estimation, we carry out extensive experiments in both synthetic and real light field datasets in Section V, respectively. Finally, Section VI concludes this paper.

II. RELATED WORKS

A. DEPTH ESTIMATION BASED ON EPIS

Since an EPI contains patterns of oriented lines and the slope of these lines is related to the depth values [3], [4], many

methods estimate depth from the LF based on EPIS, which can be roughly divided into two categories: conventional methods and deep learning methods.

Conventional methods were widely used in early studies and performed depth estimation through geometry analysis and consistency measurement. Tao et al. [5], [6] shear the EPI perform refocusing and combine the defocus and correspondence cues to produce high-quality depth estimation. Wanner and Goldluecke [7] applied the 2D structure tensor to estimate the slope of lines on EPIS. Li and Jin [8] propose a novel tensor, Kullback-Leibler Divergence (KLD), to analyze the histogram distributions of EPI's window. Then, depths calculated from vertical and horizontal EPIS' tensors are fused according to the tensors' variation scale for a high-quality depth map. Zhang et al. [9] divided an EPI into some regions using a spinning parallelogram operator (SPO) and located the lines by maximizing the distribution distance of the regions. The distance measure can keep the correct depth information, even if occluded or noisy. Subsequently, Sheng et al. [10] designed a modified SPO method that embedded SPO into multi-orientation EPIS. Johannsen et al. [11] build a dictionary for sparse light field coding and lift the trained patches to the higher dimensional epipolar space to produce a depth map. Schilling et al. [12] integrate occlusion processing into a depth model to maximize the use of the available data and obtain general accuracy and quality of object borders.

Recently, deep learning-based methods have achieved state-of-the-art performances in LF depth estimation. Heber and Pock [13] introduced deep learning technology to the application of LF depth estimation for the first time. They designed an end-to-end network to predict the depth and refined it with high-order regularization. Immediately afterward, Heber et al. [14] presented a U-shaped regression network involving two symmetric parts: encoding and decoding. This network unifies ideas from 2D EPI analysis with spatial matching-based approaches by learning 3D filters for disparity estimation based on EPI volumes. Using the idea of divide and conquer, Guo et al. [15] proposed an occlusion-aware network. This network consists of several subnetworks (such as ORDNet, CDENet, and RDENet), each of which completes a subtask of a complex task. From the perspective of LF representation, Alperovich et al. [16] designed a fully convolutional autoencoder for LF images and obtained a depth map by decoding the results of the autoencoder. Shin et al. [17] first introduced multi-stream network architecture to LF depth estimation and proposed a data augmentation method to address the issue of the lack of training data. For utilizing the texture feature of EPI, Han et al. [18] fed the synthetic EPIS and a central view image into the multi-stream network. They made a good trade-off between the efficiency and performance of the depth estimation. Tsai et al. [19] proposed an attention-based view selection network to more effectively and efficiently incorporate all angular views for depth estimation. Chen et al. [20] exploited the attention mechanism and built a multi-level fusion network to handle the occlusion

problem for depth estimation. Ma et al. [21] designed an end-to-end neural network based on atrous convolution to estimate the depth of reflective and texture-less areas. Zhou et al. [22] proposed a simple and fast cost constructor to construct matching costs for LF depth estimation.

B. RECURRENT NEURAL NETWORKS

RNN is a potent neural network model for processing and predicting sequence data. It has an excellent performance in speech recognition, machine translation, text classification, etc. The landmark achievement in this field is the long-short time memory (LSTM) network proposed by Hochreiter et al. [23], which effectively solves the problem of long-term dependence. Then, various variants, such as Gated Recurrent Unit (GRU) [24], and Bi-directional LSTM (BLSTM) [25], appeared one after another.

RNN also performs well in image and video processing. For object recognition, Visin et al. [26] proposed an RNN-based network architecture called ReNet, which uses four recurrent networks to sweep the image in both horizontal and vertical directions instead of the traditional feature extraction mode of “convolution+pooling”. Shuai et al. [27] proposed Directed Acyclic Graph-Recurrent Neural Networks (DAG-RNN) for scene segmentation. DAG-RNN aggregates context over locally connected feature maps and demonstrates noticeable performance superiority over Fully Convolution Networks (FCNs).

For depth prediction from a monocular video, Kumar et al. [28] proposed a convolutional LSTM (convLSTM)-based network architecture. Similarly, Kreuzig et al. [29] exploit a recurrent convolutional neural network to estimate the traveled distance from monocular images. The CNN part of this model is used to extract geometric features, which are subsequently input into the RNN part to learn dynamics and temporal information. Wang et al. [30] proposed an unsupervised learning model for monocular video visual odometry by using the temporal correlation properties among the input frames.

III. GEOMETRIC PRINCIPLE

A. LIGHT FIELD IMAGING AND REPRESENTATION

As shown in Fig. 1, a micro lens array (MLA) is placed between the sensor and the main lens in the LF camera. The light in different directions emitted by an object point is focused on a microlens in the MLA through the main lens, and then a micro image is captured on the sensor. The micro-image pixels record the light directions, and the microlens is the position where the light passes. Therefore, the LF camera can record the position and direction of light, while the traditional camera loses the direction information of light.

Although the raw image captured by an LF camera is two-dimensional, it can be decoded into 4D LF data and represented by the two-plane parameterization (2PP) method. 2PP method models a 4D light field as a collection of pinhole

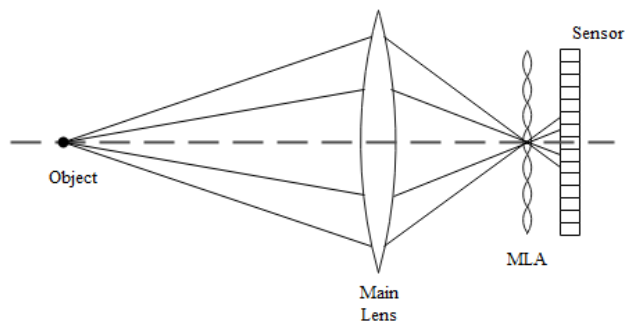


FIGURE 1. Imaging model of a light field camera.

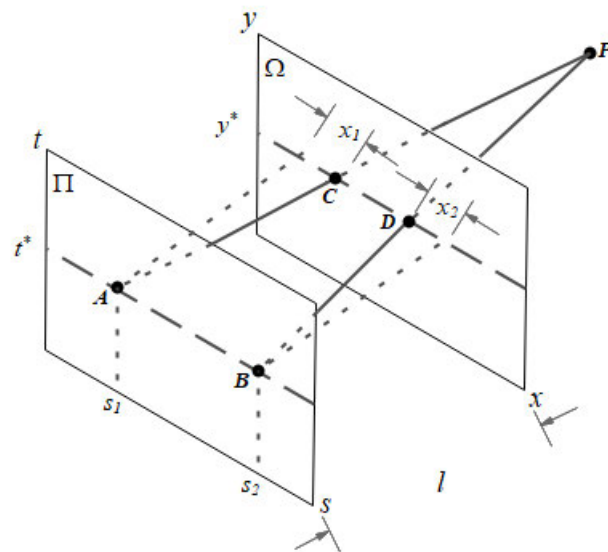


FIGURE 2. Light field representation using 2PP.

views from several viewpoints parallel to a common image plane. As shown in Fig.2, Π and Ω are two parallel planes. Π contains the viewpoints expressed in (s, t) coordinates, and Ω is the image plane parameterized by the coordinates (x, y) . Each ray not parallel to the two planes can be uniquely identified by its two intersections with the two planes. Therefore, a 4D LF can be formulated as a map from the coordinates determined by Π and Ω to the ray space R , that is

$$(x, y, s, t) \mapsto L(x, y, s, t). \tag{1}$$

In other words, this map is the process of assigning an intensity value to the ray $R_{x,y,s,t}$ passing through $(x, y) \in \Omega$ and $(s, t) \in \Pi$.

B. EPI FEATURES

In computer vision applications, it is more popular to restrict a 4D LF to a 2D slice. For the map $L(x,y,s,t)$, if the coordinates t and y are assigned to constants t^* and y^* respectively, then we get a 2D slice of the 4D LF, which is called an EPI and noted as S_{y^*,t^*} . Formally, EPI is a map from a 4D LF to a 2D image, and can be described as

$$S_{y^*,t^*} : (x, s) \mapsto L(x, y^*, s, t^*). \tag{2}$$



FIGURE 3. EPI examples. (a) is a central view of a light field, and (b) and (c) are EPIs corresponding to the positions of the blue vertical line and the red horizontal line in the central view, respectively.

Similarly, if the coordinates s and t are set to s^* and t^* , respectively, that is, the viewpoint fixed at the point (s^*, t^*) on the plane Π , we can observe a view image I_{s^*, t^*} , also known as the sub-aperture image. If (s^*, t^*) is the center of all viewpoints, I_{s^*, t^*} is named the central view (CV). Here I_{s^*, t^*} can be formalized as

$$I_{s^*, t^*} : (x, y) \mapsto L(x, y, s^*, t^*). \quad (3)$$

For visualization, Fig.3 illustrates an example of a central view and two EPIs, where (a) is a central view of a light field, and (b) and (c) are EPIs corresponding to the positions of the blue vertical line and the red horizontal line in (a), respectively. EPIs in the example show obvious linear texture features. Behind this phenomenon is the geometric principle of LF imaging. Let us review the geometry of Fig.2, a point P in the epipolar plane with the invariant coordinate values of y^* and t^* projects different image points (C or D) depending on the chosen viewpoints (A or B), respectively. According to the triangular similarity, the relationship between the viewpoint shifts (Δs) and the change (Δx) of the image point can be formulated as Equation (4). Here $\Delta s = s_2 - s_1$, $\Delta x = x_2 - x_1$; Z represents the depth of the point P , and l denotes the distance between two planes, Π and Ω .

$$\frac{\Delta s}{\Delta x} = -\frac{Z}{l} \quad (4)$$

As a result, the pixels corresponding to point P have similar gray levels and are arranged in a straight line in EPI, and the slope of the straight line is related to the depth of point P .

To further analyze the texture features of EPI, we enlarge the EPI and draw a schematic diagram (Fig. 4). Each square marked with r , g , b or p in the figure represents a pixel, and each letter indicates the gray value of the pixel. It is evident that a straight line in EPI is composed of several pixels. For instance, the pixels in the green area in Fig. 4 are arranged in a straight line (Lg).

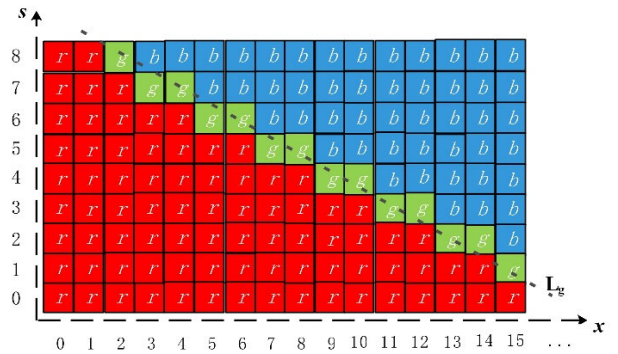


FIGURE 4. Schematic diagram of vector sequences in EPI. Each small grid corresponds to a pixel, with the letters 'r', 'g', and 'b' indicating its grayscale value. The pixels labeled with 'g' are used to fit a straight line, represented as Lg.

In Fig.4, if we regard each column pixel of EPI as a vector, e.g., the 0th and 3rd columns are expressed as (r, r, r, r, r, r, r, r) and (b, g, r, r, r, r, r, r) respectively, then the straight line Lg is included in the vector sequence from the second to 15th columns. Due to the existence of Lg, there is a certain correlation among these 14 vectors. Based on this observation, it may be a feasible scheme to make good use of a neural network to learn the correlation features in this vector sequence and predict the slope of the straight line or the corresponding depth.

IV. METHODOLOGY

We begin by utilizing CRF to model the process of estimating depth in light fields. Building upon this, we design a novel deep network that leverages the principles of geometry and observations to accurately estimate depth. The unique framework is shown in Fig. 5 and explained in the subsequent sections, i.e., local depth estimation and depth refinement. As the disparity value of a pixel is inversely proportional to the depth value, our LF depth estimation model's final output is the LF image's disparity map.

A. FORMULATION FOR DEPTH ESTIMATION

The method for LF depth estimation involves obtaining a disparity map for the central view through statistical analysis of light field data. Markov Random Field (MRF) or CRF models have been commonly applied to solving dense prediction problems, such as monocular depth estimation and image semantic segmentation. In this section, we introduce CRF into the depth estimation of light field.

Consider the central view image (CV) as a random field defining over a set of variables $\{I_1, I_2, \dots, I_N\}$. The disparity map associated with the CV can also be defined a random field over a set of variables $\{d_1, d_2, \dots, d_N\}$. I_j is the color vector of pixel j and d_j is the disparity value assigned to pixel j . N indicates the number of pixels in the CV. The pair (\mathbf{I}, \mathbf{d}) can be modeled as a CRF characterized by a Gibbs

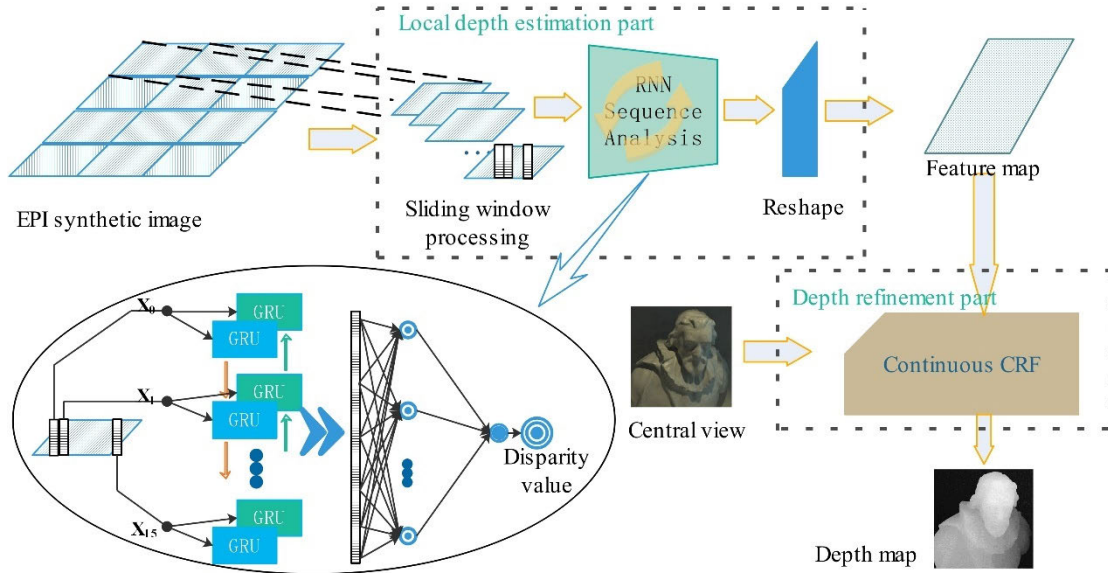


FIGURE 5. Our network architecture. We call this network LFRNN. It takes the conditional random field as the theoretical model, and achieves accurate depth estimation through local depth estimation and depth refinement.

distribution of the form

$$P(\mathbf{d}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp\{-E(\mathbf{d}, \mathbf{I})\}, \quad (5)$$

where $Z(\mathbf{I}) = \int_{\mathbf{d}} \exp\{-E(\mathbf{d}, \mathbf{I})\} d\mathbf{d}$ is a partition function, and the energy function $E(\mathbf{d}, \mathbf{I})$ is defined as Eq. (6). Generally, the energy function consists of two components: the unary energy component and the pairwise energy component. The unary energy component $\psi_u(d_i)$ is computed independently for each pixel by the classifier, described as local depth estimation. Furthermore, pairwise energy component $\psi_p(d_i, d_j)$ measures the cost of assigning disparity values d_i and d_j to pixels i and j , respectively.

$$E(\mathbf{d}, \mathbf{I}) = \sum_i \psi_u(d_i) + \sum_{i < j} \psi_p(d_i, d_j) \quad (6)$$

We continue to define the unary energy component $\psi_u(d_i)$ and the pairwise energy component $\psi_p(d_i, d_j)$ as shown in Eqs. (7) and (8), respectively. In Eq. (7), o_i is the regress disparity value at pixel i . In Eq. (8), $w_m(i, j, \mathbf{I})$ is a weight that specifies the relationship between the estimated disparity of the pixels i and j ; M is the number of kernels.

$$\psi_u(d_i) = (d_i - o_i)^2 \quad (7)$$

$$\psi_p(d_i, d_j) = \sum_{m=1}^M \beta_m w_m(i, j, \mathbf{I})(d_i - d_j)^2 \quad (8)$$

As mentioned above, the unary potential component measures the inverse likelihood of the pixel j take the disparity d_j . The current methods compute the unary potential component for each pixel by a deep network predictor according to the image features such as shape, texture, location, color and so on. Considering the rich information contained in the light field data, we construct a unary component based on EPI synthetic image instead of a simple central view. The pairwise

component provide an image data-dependent smoothing term that encourages assigning similar disparity values to pixels with similar properties. We will describe the implementation of Eq. (8) in the depth refinement subsection.

B. NETWORK ARCHITECTURE

In general, our network includes two parts: a local depth estimation part and a refinement part. The local depth estimation takes the EPI synthetic image as the input and outputs the local disparity map. The EPI synthetic image is composed of EPI corresponding to each row of the CV. Concretely, EPIs are first generated from the LF data for each row of pixels in CV (see Fig.3), and then those EPIs are spliced from top to bottom to form an EPI synthetic image, as shown in the input image diagram in Fig.5.

After receiving the EPI synthetic image, the local depth estimation part obtains a local disparity map through the sliding window processing layer, sequence analysis module, and reshape layer. The sliding window processing layer is similar to the sliding process of the convolution kernel in CNNs. This layer's purpose is to provide the EPI patches for the sequence analysis layer and facilitate parallel processing. The sequence analysis module is the key to extracting local features from each EPI patch, which will be expanded in detail in subsequent subsection. Next, we transform the output of the sequence analysis layer into a two-dimensional matrix (feature map) with the exact resolution as the CV. This feature map is used as the initial disparity map and then input to the refinement part.

The refinement part is to optimize the initial disparity map based on the spatial and color information of the CV. We utilize the CRF theory to model this optimization problem. Therefore, this network module has two inputs, one is the feature map from the local depth estimation part, and the other

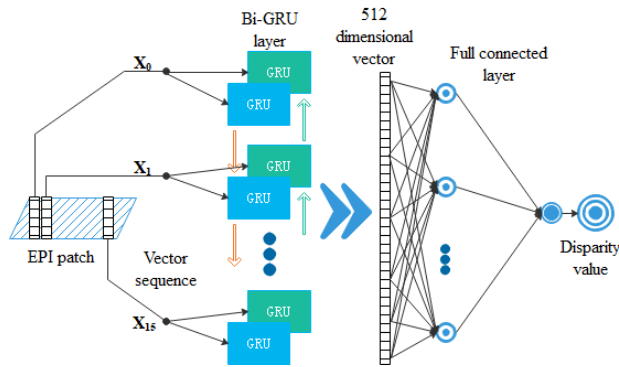


FIGURE 6. Illustration of sequence analysis module. The module first uses bidirectional GRU to extract the sequence features, and then regresses the disparity value through the full connection layer.

is the CV. The feature map is used as the unary term of the CRF model. Moreover, the pairwise term of the CRF model takes into account the pixel position and color information of the CV.

C. LOCAL DEPTH ESTIMATION

As shown in Fig.5, the local depth estimation includes three components: sliding window processing, sequence analysis module, and reshape layer. This subsection focuses on the sequence analysis module. Section III has described an important observation: the pixels of a straight line in EPI are arranged in vector sequences, and the slope of the straight line determines the relationship between the vector sequences. Moreover, RNN has been successfully applied in sequence analysis [28], [29]. Therefore, we exploit RNN to design a sequence analysis module suitable for local depth estimation, as illustrated in Fig.6.

In Fig. 6, Gated Recurrent Unit (GRU) is a typical unit of the RNN, which has similar performance to the unit of the Long Short-Term Memory (LSTM) network but is much simpler to compute and implement. Fig.7 shows the structure of a GRU cell. There exist two kinds of gates, i.e., reset gate r_t and update gate z_t . The reset gate is used to combine the new input with the previous memory, and the update gate checks how much of the information from the previous state flows to the current hidden state. The general equations of GRU cell are shown in Eqs. (9)-(12). Herein, x_t represents the layer's input as an m -dimensional vector, and h_{t-1} is the hidden state of the time step of $t-1$. \tilde{h}_t and h_t are the candidate hidden state and the new state, respectively. W_r , W_z , and W_h are weight parameters. $\sigma()$ and $\tanh()$ denote sigmoid and hyperbolic tangent functions, respectively.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (9)$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (10)$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \times h_{t-1}, x_t]) \quad (11)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (12)$$

As shown in Fig.6. Each column of the EPI patch is regarded as a vector and input into the network. After analyzing the vector sequence through the bidirectional GRU layer

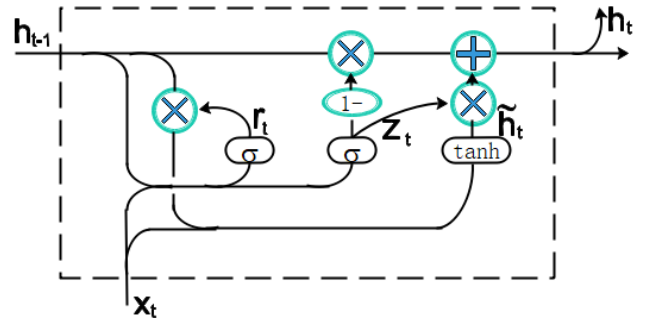


FIGURE 7. Structure of GRU cell. GRU associates the current input with the historical state through some gates to form a new output.

and full connection layers, the disparity value corresponding to the EPI patch is obtained. In practice, we set the EPI patch size to 9×16 and divide the EPI patch into 16 9-dimensional vectors.

Considering that the sequence feature is related to the forward and backward vectors at the analysis point, we design a bidirectional RNN with GRU cells as the hidden state layer of the sequence analysis module. Then we flatten the output of bidirectional GRUs into a 1-dimensional vector as the input of the full connection layers. The bi-GRU layer consists of two directional GRU cells with a dimension of 256 in each direction, and each GRU cell is set to a non-sequential mode of operation, which receives 16 vector inputs and produces an output value. In total, the bi-GRU layer generates 512 output values.

The next components of the sequence analysis module are the two full-connection layers. The first full-connection layer achieves a nonlinear transform from 512 to 16 dimensions through the activation function ReLU. In contrast, the second full-connection layer without activation function completes the mapping of 16-dimensional vectors to a disparity value.

D. DEPTH REFINEMENT

So far, we have predicted disparity values for pixels using regression. Now our goal is to refine the predicted disparity values from EPI patches.

In order to quickly solve the fully connected CRFs model, Krähenbühl and Koltun [31] defined the pairwise energy component as a linear combination of Gaussian kernels. They reduced the computational complexity of message passing from quadratic to linear in the number of variables by employing efficient approximate high-dimensional filtering. In recent years, some schemes based on a neural network to solve the CRF model have appeared. Zheng et al. [32] interpreted dense CRFs as RNNs and indicated that the parameters of CRFs can be learned during the backward propagation process of training the deep neural network. Based on these works, Xu et al. [33] utilized a clever approach of extracting multi-scale features from a single image and then fusing them using a CRF. In our method, we employ the sequence features obtained from the local depth estimation module as a unary potential component within the CRF. Nevertheless,

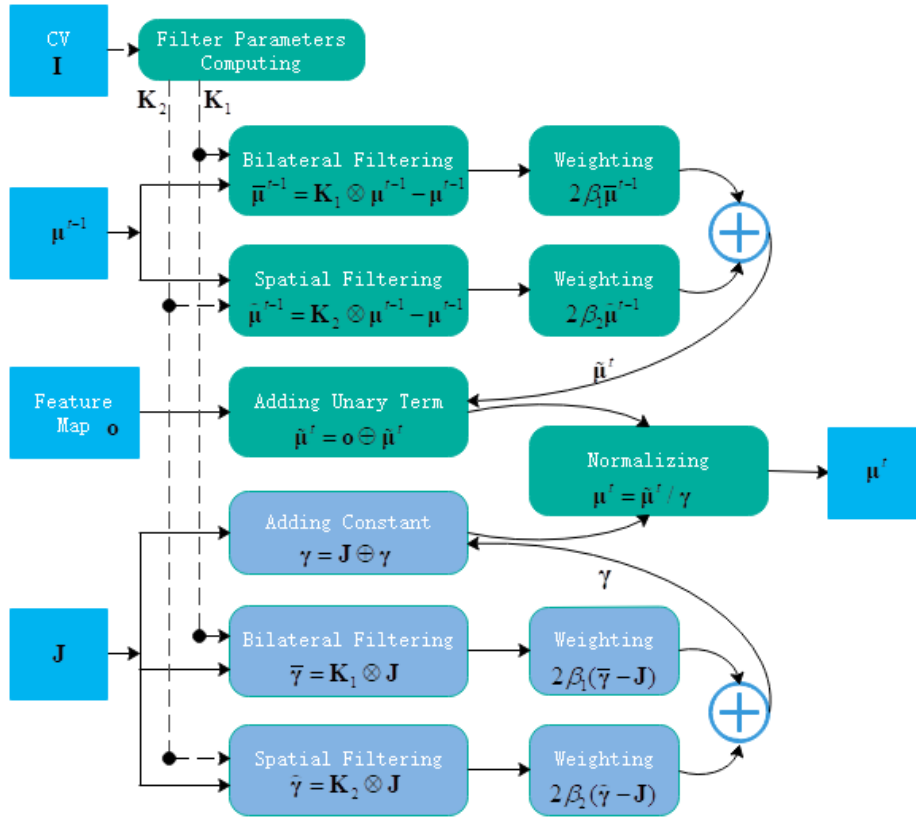


FIGURE 8. Iterative process of depth refinement.

Xu's C-MF module presents valuable insights and is worth considering for further application. Following the work of Xu et al. [33], we use Eqs. (13) and (14) to update the variance and mean of the approximate distribution, respectively.

$$\gamma_i = 2(1 + 2 \sum_{m=1}^M \beta_m \sum_{j,i} w_m(i, j, \mathbf{I})) \quad (13)$$

$$\mu_i = \frac{2}{\gamma_i} (o_i + 2 \sum_{m=1}^M \beta_m \sum_{j,i} w_m(i, j, \mathbf{I}) \mu_j) \quad (14)$$

We also use two features proposed by [31] to define the weights $w_m(i, j, \mathbf{I})$ as follows:

$$w_1(i, j, \mathbf{I}) = \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right), \quad (15)$$

$$w_2(i, j, \mathbf{I}) = \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right). \quad (16)$$

here, I_i and I_j are the color vectors of pixels i and j , and p_i and p_j are the positions associated with pixels i and j . θ_α , θ_β and θ_γ are user-defined bandwidth parameters. $w_1(i, j, \mathbf{I})$ is usually called appearance kernel, which takes into account the prior knowledge that nearby pixels with similar colors are likely to be of a similar depth. $w_2(i, j, \mathbf{I})$ is also named smoothness kernel because it can filter out some outliers.

As for the iterative process of solving the CRF, our main idea is to modify the multi-scale continuous mean field

(C-MF) in [33] to a single-scale mean field. Our iterative process is shown in Fig. 8. Inputs of iteration t include the output μ^{t-1} at iteration $t-1$, the central view image \mathbf{I} , the output feature map o of the local depth estimation network, and an identity matrix J . First, we use Eqs. (15) and (16) to calculate w_1 and w_2 of each pixel of the CV and then form kernel matrices \mathbf{K}_1 and \mathbf{K}_2 . On the two input paths, μ^{t-1} and J , the kernel matrices \mathbf{K}_1 and \mathbf{K}_2 are used for bilateral and spatial filtering. And then, the unary term is added to the processing result of path μ^{t-1} . At the same time, J is superimposed on that of the path J . Finally, the two processing results on the two paths are divided element by element to get the output of this iteration.

The processing depicted in Fig. 8 can be interpreted as a single-scale C-MF module. By interconnecting multiple such modules with shared parameters, we construct an RNN for solving the CRF. That is, the output μ^t at iteration t is used as the input at iteration $t+1$. This way, an RNN similar to [32] is built, but our network outputs are continuous disparity values.

Usually, we take the feature map captured by the local depth estimation network as the initial value of μ^{t-1} at the first iteration. No more than 10 iterations can output the ideal result in the experiment. Our network training goes through two stages. The first stage is to train the local depth estimation network by using the ground truth value of disparity, and the whole network is trained in the second stage. In order to improve the training efficiency, the parameters of the local

depth estimation network can be locked in the second stage, and only the weighted parameters in the iterative optimization network are trained.

V. EXPERIMENTS AND ANALYSIS

A. IMPLEMENTATION

This paper uses the HCI dataset train and test our network model LFRNN [34]. The dataset includes 28 scenes and is divided into four categories: *structured*, *test*, *training*, and *additional*. The dataset designer fully considered common visual problems such as materials, lighting conditions, and occlusion. They used Blender software to render the LF images with a spatial resolution of 512×512 and angular resolution of 9×9 . Like most current LF depth estimation methods, we use 16 scenes in *additional* for training, and 12 scenes in *structured*, *test* and *training* for verification and testing.

We implement the proposed LFRNN network based on the Pytorch framework and the i9 CPU and P100 GPU platform. First, we generate the CV image and EPI synthetical image from the 4D LF data of the scene and input them into our LFRNN. Second, using the idea of space for time, we exploit the Unfold layer in Pytorch to convert EPI synthetical images into EPI patches analyzed by the local depth estimation module. Moreover, the CRF module follows the internal iteration implementation methods of [32] and [33], and the number of iterations is empirically set to 6.

Our LFRNN is trained in two stages. In the first stage, the local depth estimation part is regarded as an independent neural network with the EPI synthetical image as the input and the feature map as the output. Thus, the training is conducted under the supervision of the ground truth provided by the HCI dataset. Our loss function is the Mean Absolute Error (MAE), and the learning rate of the first 1000 epochs is set to 10^{-3} and then to 10^{-4} . After each epoch, the model effect is evaluated on the verification set to decide whether to preserve the current model parameters. We use Visdom to visualize the training process and end the local depth estimation network training when the loss function is stable. The second stage is to train the CRF optimization part. We freeze the parameters of the local depth estimation part so that the whole LFRNN forms an end-to-end network. β_1 and β_2 are the key parameters to learn at this stage. After 6 rounds of iteration in the experiment, the network can achieve stable performance.

B. EVALUATION METRICS

In the training process, we employ MAE, that is, L1 loss in Pytorch, to evaluate the difference between the currently estimated disparity and the ground truth. Although the convergence speed of MAE is slow, it can better reflect the actual situation of prediction error and is suitable for synthetic datasets with relatively few outliers.

In comparison with similar methods, we adopt the widely used evaluation standards recommended by the HCI dataset, namely MSE and Badpix. For the sake of clarity, we report

this as follows. Given an estimated disparity map \mathbf{d} , the ground truth disparity map \mathbf{gt} and an evaluation mask \mathbf{M} , MSE and Badpix are defined as Eqs. (17) and (18), respectively. In (18), t is a disparity error threshold, usually set to one value of 0.01, 0.03, and 0.07.

$$\text{MSE}_{\mathbf{M}} = \frac{\sum_{x \in \mathbf{M}} (\mathbf{d}(x) - \mathbf{gt}(x))^2}{|\mathbf{M}|} \times 100 \quad (17)$$

$$\text{BadPix}_{\mathbf{M}}(t) = \frac{|\{x \in \mathbf{M} : |\mathbf{d}(x) - \mathbf{gt}(x)| > t\}|}{|\mathbf{M}|} \quad (18)$$

It can be seen from Eqs (17) and (18) that MSE measures the average error of the disparity values, and BadPix reflects the proportion of pixels that reach the error threshold. In the following experimental evaluation, we set mask \mathbf{M} as a single pixel area to see the disparity error at each pixel and observe the method's robustness to occlusion.

C. ABLATION INVESTIGATION

As previously mentioned, the first half of the LFRNN network is local depth estimation, and the second half is the CRF optimization module. The former can independently estimate the disparity, while the latter is optimizing the former results. In order to verify the effectiveness of the CRF module, we designed an ablation experiment, that is, to compare the estimation results of the local depth estimation module with that of the whole network LFRNN.

We call the local depth estimation part LDE and the whole network LFRNN. The two networks are tested on *Cotton*, *Dino*, *Pyramids*, and *Stripes*. According to the ground truth provided by the HCI dataset, we calculate the MSE and Badpix metrics obtained by the two networks in each scene. As shown in Table 1, the test results of LFRNN in all scenes are better than those of LDE, indicating that the CRF optimization module can indeed improve the accuracy of depth estimation. Taking the scenes of *Cotton* and *Pyramids* as examples, Fig. 9 intuitively shows the depth estimation results of LDE and LFRNN, where the first column is the two central views, and the second and third columns are the disparity maps of LDE and LFRNN respectively. LFRNN's result is smoother in the forehead area of *Cotton* and more explicit in the boundaries of the *Cotton*'s right ear and cape than those in LDE's result. On the other hand, The *Pyramids* scene is specifically created to evaluate the algorithm's performance in handling slanted, convex, or concave scenes. When comparing the results of the LFRNN with LDE, it is observed that LFRNN produces sharper object boundaries. This means that the disparities between objects are more precisely estimated, resulting in a clearer distinction between different objects in the scene. Additionally, LFRNN demonstrates an improved ability to capture depth changes on convex or concave surfaces. Disparity values on these surfaces are estimated more accurately, and the transitions from one depth level to another are more clearly defined. The above phenomena may be because CRF module adjusts the local disparity estimation from a global perspective.

TABLE 1. Results of ablation experiment.

Methods	Cotton		Dino		Pyramids		Stripes	
	MSE	Badpix	MSE	Badpix	MSE	Badpix	MSE	Badpix
LDE	0.387	0.409	0.112	0.986	0.011	0.203	0.914	3.126
LFRNN	0.245	0.343	0.093	0.898	0.006	0.189	0.875	2.997

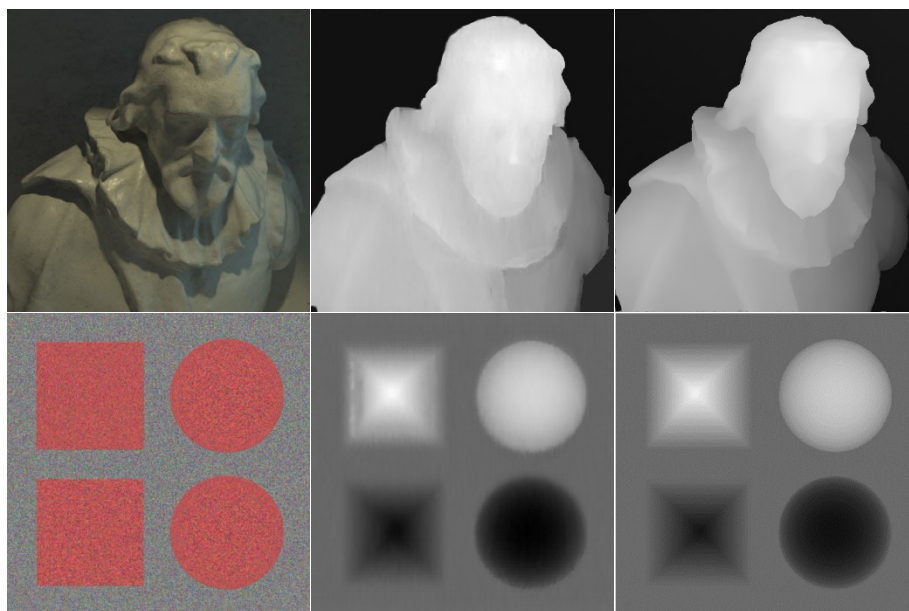


FIGURE 9. Examples of ablation results. The first column is the central view image of the scene Cotton, the second column is the disparity map estimated by LDE, the third column is the disparity map estimated by LFRNN.

D. SYNTHETIC DATASET

We have submitted the experimental results of the LFRNN to the HCI benchmark for testing. Currently, the benchmark evaluates a total of 115 methods, and our LFRNN ranks 17th and 15th in terms of median and average Badpix metrics, respectively. However, when considering the MSE metrics, our LFRNN achieves first place in terms of median and ninth place in terms of average.

Table 2 presents the results of LFRNN along with four other methods: EPINet [17], VommaNet [21], and FusionNet [22]. The evaluation includes 12 scenes, out of which 8 scenes have publicly disclosed disparity ground truth, while 4 scenes have unpublished disparity ground truth. In the table, the bold numbers indicate the best MSE or Badpix values among the five methods for a particular scene. It is worth noting that the LFRNN consistently achieves optimal values in most scenes, indicating its strong performance compared to the other methods. Overall, the LFRNN algorithm demonstrates favorable results in terms of both MSE and Badpix metrics across the majority of the evaluated scenes in the HCI benchmark.

In order to visualize the metric results of each method, we have drawn a percentage radar chart of the experimental results. Fig. 10 shows the percentage radar chart of

these methods in eight scenes, such as *Boxes*, *Cotton*, *Dino*, *Sideboard*, *Backgammon*, *Dots*, *Pyramids*, and *Stripes*. Each radial axis in the figure corresponds to a scene, and each method is marked as a point on the axis. Among them, the method with the maximum MSE value is marked as 100%, and other methods draw points according to the percentage of their MSE values in the maximum MSE value. In this way, the points marked by the same method on each axis are connected to form a polygon. The polygon is in the inner or outer layer, indicating the advantages and disadvantages of the method. Similarly, the percentage value distribution of these methods regarding the Badpix metric is illustrated in Fig. 11.

As shown in Fig. 10, the MSE polygon of LFRNN is almost at the innermost level and has the smallest area, so the MSE average of LFRNN is the smallest in all scenes. Specifically, LFRNN achieves optimal MSE values in almost all scenes except *Dots* and *Backgammon*. Although LFRNN’s MSE is not as good as VommaNet’s in *Dots* and *Backgammon* scenes, LFRNN’s BadPix is much better than VommaNet’s. Overall, LFRNN’s BadPix metrics also perform exceptionally, showing optimal or sub-optimal performance in all scenes. Combined with Fig. 10 and Fig. 11, LFRNN has the optimum values of the two metrics for complex occlusion scenes *Boxes*, *Cotton*, *Dino*, and *Sideboard*, which shows that our method

TABLE 2. Results of relevant methods on the hci synthetic dataset.

Scenes	LFRNN		FUSIONNET		EPINET		VOMMANET	
	MSE	Badpix	MSE	Badpix	MSE	Badpix	MSE	Badpix
Backgammon	3.581	3.206	5.148	3.797	3.909	3.287	3.532	7.709
Dots	1.559	1.860	2.418	1.742	1.980	4.030	1.143	9.136
Pyramids	0.006	0.189	0.009	0.296	0.007	0.147	0.026	0.588
Stripes	0.875	2.997	1.974	4.026	0.915	2.413	1.118	9.289
Bedroom	0.093	2.188	0.460	3.629	0.231	2.287	0.331	7.371
Bicycle	3.150	9.454	6.214	9.652	4.929	9.853	4.960	19.175
Herbs	6.474	5.275	10.561	7.225	9.423	17.756	12.322	27.391
Origami	0.832	3.992	4.211	5.672	1.646	6.339	1.458	11.676
Boxes	2.825	10.886	7.897	11.816	6.036	12.248	4.304	20.641
Cotton	0.245	0.343	0.451	0.551	0.223	0.464	0.289	2.620
Dino	0.093	0.898	0.678	2.451	0.151	1.263	0.317	4.933
Sideboard	0.545	3.038	1.556	5.235	0.806	4.783	0.875	9.652

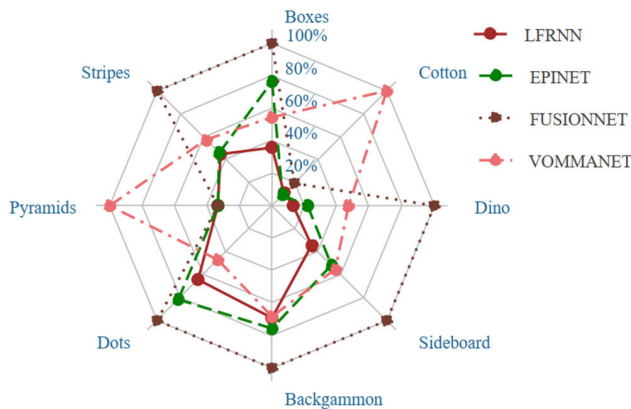


FIGURE 10. Percentage radar chart of MSE metric.

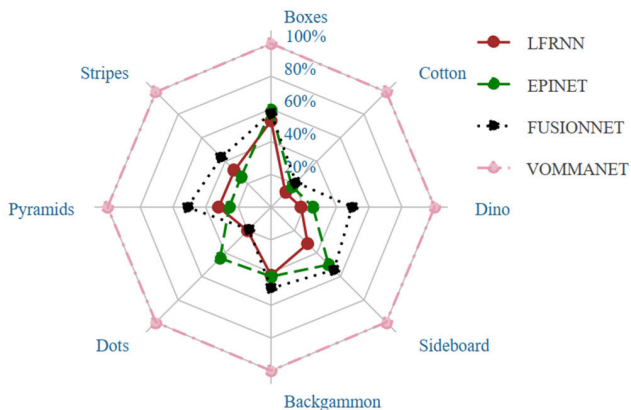


FIGURE 11. Percentage radar chart of Badpix metric.

has better average performance and has some advantages in occlusion processing.

Fig. 12 shows the CVs of the scenes of *Boxes*, *Dino* and *Backgammon*. On the other hand, Fig. 13 illustrates the

pixel-by-pixel distribution of MSE and Badpix obtained by the respective methods on the three scenes. Each column in Fig. 13 corresponds to the results of a method: LFRNN, EPINet, FusionNet, and WommaNet. Each scene is presented in two rows, showcasing the errors of Badpix and MSE metrics separately. Color rulers are attached at the end of each row. The Badpix ruler uses red and green colors. Specifically, the pixels with small Badpix errors are represented in green, but red means large Badpix. The MSE metric charts show positive deviation in red and negative deviation in blue. The lighter the color, the smaller the MSE. The number above each image represents the overall score of the metric.

On the whole, LFRNN achieves the minimum Badpix values in both scenes compared with other methods. From the Badpix distribution map (Fig. 13), the pixels in the smooth area are all green, and the individual occlusion boundary is red. Especially in the case of complex occlusion in the scene *Boxes*, the red area of the LFRNN map is relatively sparse, indicating that LFRNN has good performance in dealing with occlusion.

For MSE metrics, Vommanet achieved the best results in the scene of Backgammon. However, from the perspective of error distribution, its results show red and green in many areas; that is to say, there is a large area of positive and negative deviation, which offsets the overall MSE. On the contrary, LFRNN’s result is light in most areas, which means the absolute error is small.

E. REAL-WORLD DATASET

We conducted experiments on two real LF datasets, namely Lytro [35] and LytroIllum [36], respectively. Since the real LF datasets do not include depth truth values, we only make qualitative comparisons. According to the availability of the implementation code, we choose SPO [9], EPINet [17],

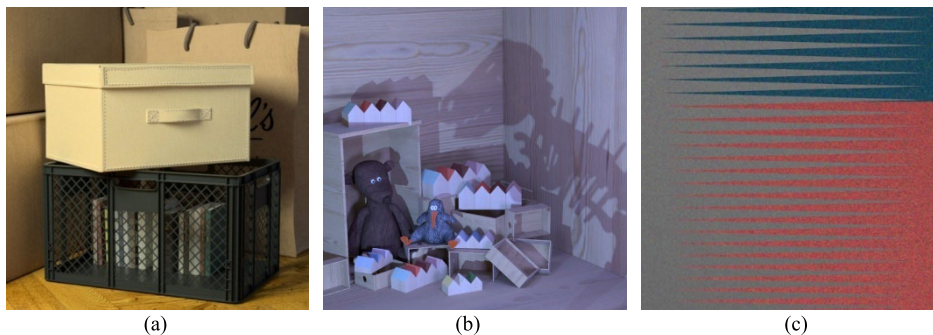


FIGURE 12. Central view images of three scenes. (a), (b) and (c) are the central view images of the scenes of Boxes, Dino and Backgammon, respectively.

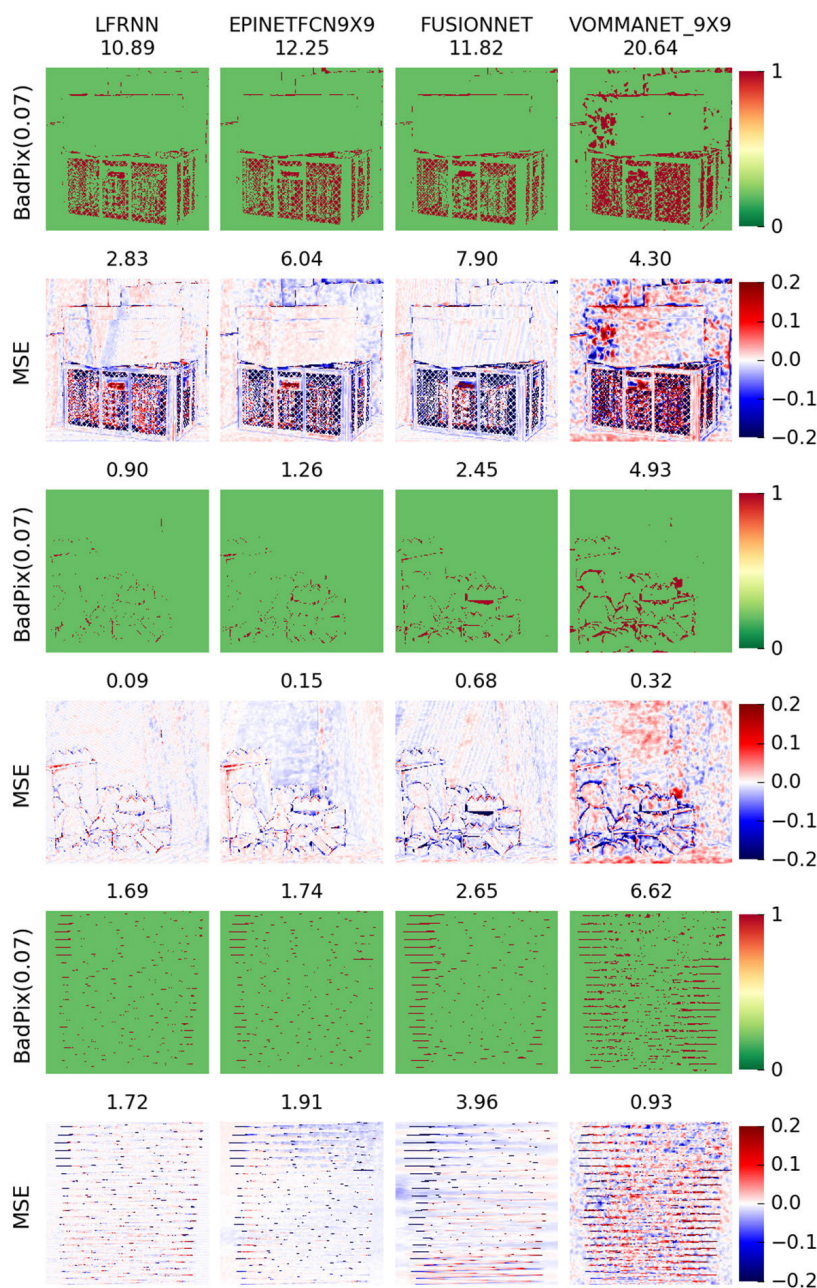


FIGURE 13. Evaluation examples on the synthetic dataset.

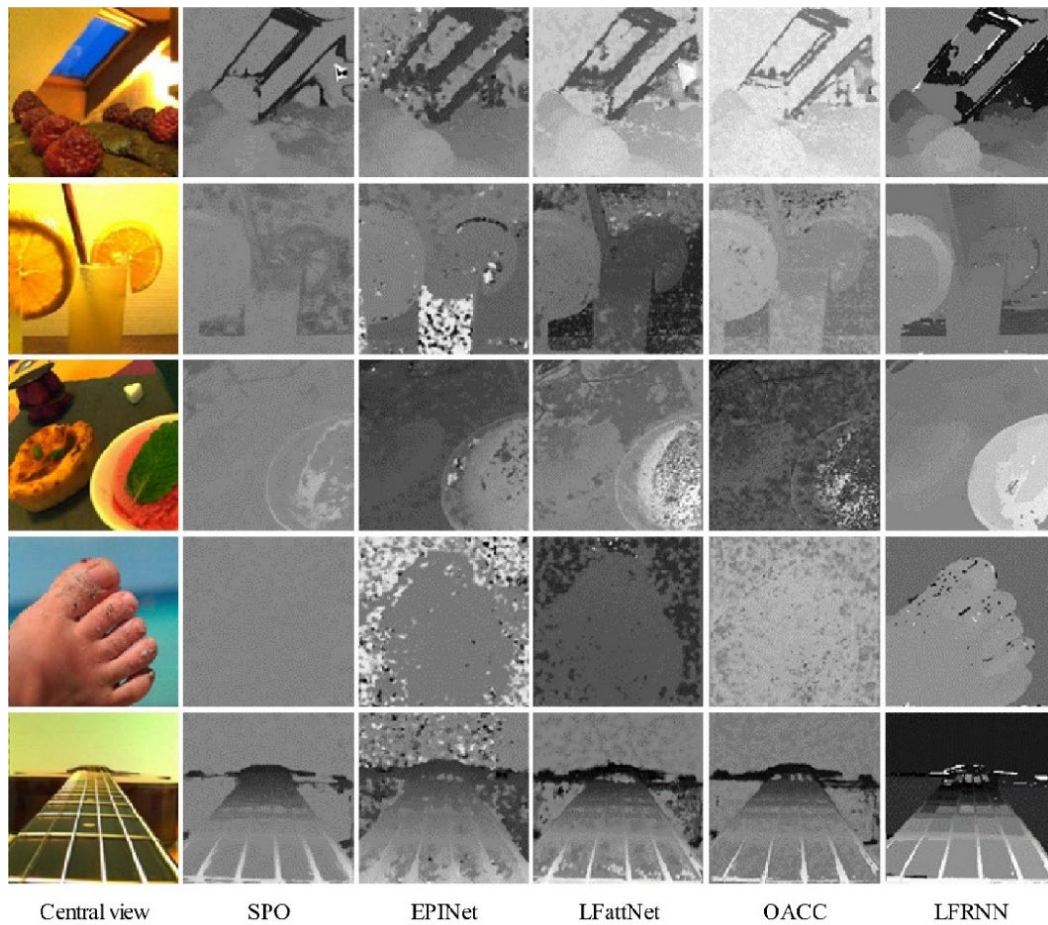


FIGURE 14. Examples of disparity results on Lytro dataset.

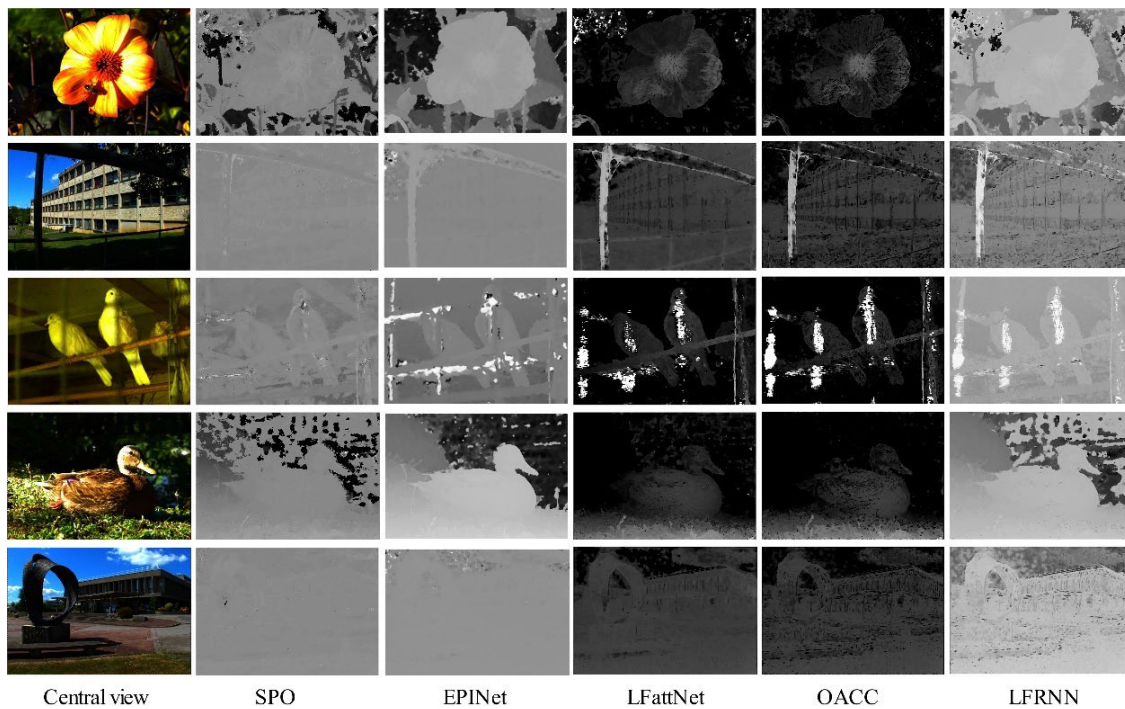


FIGURE 15. Examples of disparity results on Lytroillum dataset.

LFattNet [19], and OACC [2] methods for comparative analysis. Among these methods, SPO [9] is a traditional method; EPINet [17], LFattNet [19], OACC [2] and our LFRNN are learning-based methods. Fig. 14 shows some examples of the disparity results on Lytro dataset. The first column represents the CVs of the testing scenes, and the following columns are the disparity map obtained by the methods SPO [9], EPINet [17], LFattNet [19], OACC [2] and our LFRNN in turn. From top to bottom, the scenes corresponding to each row are *Cake*, *Cocktails*, *Dessert*, *Flat_toes*, and *Guitar*.

From the example scenes in Fig. 14, the disparity noise and rollover phenomena appear in the results of those methods. For example, SPO has some disparity noise in the scenes of *Cake* and *Guitar* and almost no perception of the disparity level in *Flat_toes*; EPINet has the most significant disparity noise and poor smoothness; LFattNet shows disparity rollover phenomena in *Cake* and *Guitar* scenes; OACC improves disparity smoothness, but performs poorly in the scene of *Dessert*. Relatively speaking, our LFRNN achieves the best disparity hierarchy and smoothness.

Similarly, Fig. 15 showcases examples of the results obtained from the five aforementioned methods in five typical scenes (*Bee_2*, *Building*, *Doves*, *Duck*, *Sculpture*) of the LytroIllum dataset. SPO and EPINet demonstrate good performance in slightly closer scenes, specifically *Bee_2*, *Doves*, and *Duck*. However, in slightly farther scenes like *Building* and *Sculpture*, the discernible hierarchy in their disparity maps diminishes. On the other hand, LFattNet and OACC exhibit better performance in slightly distant scenes. Notably, our LFRNN achieves relatively ideal disparity results in both of these scene categories.

On real LF datasets, all five methods were run on the same workstation, allowing for a fair comparison of their computational time. The average computational time was calculated for each method in different scenes using the two datasets, and the results are as follows:

SPO exhibited the longest average computational time, taking approximately 1100 seconds. This longer duration can be attributed to its implementation in Matlab, which lacks GPU utilization during the running process. LFattNet, utilizing 3D convolution that involves extensive computations, had an average computational time of 3.352 seconds. EPINet, OACC, and LFRNN, on the other hand, exhibited average computational time of 0.927 seconds, 0.034 seconds, and 0.026 seconds, respectively. It is evident that the computational time of LFRNN and OACC are quite similar, likely due to the clear and straightforward nature of their models, which enables efficient computations.

VI. CONCLUSION

EPI is a kind of computational image of LF data. The slope of the straight line in EPI contains the depth information of the corresponding object points, which has been widely used in LF depth estimation. We further point out that a line in EPI is composed of vector sequences, and the slope of the line affects the sequence distribution. Given this

observation, we propose a depth estimation network LFRNN based on sequence analysis, including local depth estimation and depth refinement. In the first part, RNN analyzes the vector sequence of EPI patches and gets a local disparity map. The second part optimizes the results of the first part based on CRF theory and outputs the final disparity map. We train LFRNN on the synthetic dataset and test it on the synthetic and real LF datasets, respectively. The experimental results show that LFRNN achieves good performance. Our research work verifies the feasibility of sequence analysis and provides a new idea for LF depth estimation. In the future, we will optimize the network and continue to improve the accuracy of depth estimation.

REFERENCES

- [1] K. Han, W. Xiang, E. Wang, and T. Huang, "A novel occlusion-aware vote cost for light field depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8022–8035, Nov. 2022, doi: [10.1109/TPAMI.2021.3105523](https://doi.org/10.1109/TPAMI.2021.3105523).
- [2] Y. Wang, L. Wang, Z. Liang, J. Yang, W. An, and Y. Guo, "Occlusion-aware cost constructor for light field depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 19777–19786, doi: [10.1109/CVPR52688.2022.01919](https://doi.org/10.1109/CVPR52688.2022.01919).
- [3] C. Perwass and L. Wietzke, "Single lens 3D-camera with extended depth-of-field," *Proc. SPIE*, vol. 8291, Feb. 2012, Art. no. 829108.
- [4] H. Lv, K. Gu, Y. Zhang, and Q. Dai, "Light field depth estimation exploiting linear structure in EPI," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1–6, doi: [10.1109/ICMEW.2015.7169836](https://doi.org/10.1109/ICMEW.2015.7169836).
- [5] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 673–680, doi: [10.1109/ICCV.2013.89](https://doi.org/10.1109/ICCV.2013.89).
- [6] M. W. Tao, P. P. Srinivasan, S. Hadap, S. Rusinkiewicz, J. Malik, and R. Ramamoorthi, "Shape estimation from shading, defocus, and correspondence using light-field angular coherence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 546–560, Mar. 2017, doi: [10.1109/TPAMI.2016.2554121](https://doi.org/10.1109/TPAMI.2016.2554121).
- [7] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4D light fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 41–48, doi: [10.1109/CVPR.2012.6247656](https://doi.org/10.1109/CVPR.2012.6247656).
- [8] J. Li and X. Jin, "EPI-neighborhood distribution based light field depth estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2003–2007, doi: [10.1109/ICASSP40776.2020.9053664](https://doi.org/10.1109/ICASSP40776.2020.9053664).
- [9] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Comput. Vis. Image Understand.*, vol. 145, pp. 148–159, Apr. 2016.
- [10] H. Sheng, P. Zhao, S. Zhang, J. Zhang, and D. Yang, "Occlusion-aware depth estimation for light field using multi-orientation EPIs," *Pattern Recognit.*, vol. 74, pp. 587–599, Feb. 2018.
- [11] O. Johannsen, A. Sulc, and B. Goldluecke, "What sparse light field coding reveals about scene structure," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3262–3270, doi: [10.1109/CVPR.2016.355](https://doi.org/10.1109/CVPR.2016.355).
- [12] H. Schilling, M. Diebold, C. Rother, and B. Jähne, "Trust your model: Light field depth estimation with inline occlusion handling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4530–4538.
- [13] S. Heber and T. Pock, "Convolutional networks for shape from light field," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3746–3754.
- [14] S. Heber, W. Yu, and T. Pock, "Neural EPI-volume networks for shape from light field," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2271–2279.
- [15] C. Guo, J. Jin, J. Hou, and J. Chen, "Accurate light field depth estimation via an occlusion-aware network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6, doi: [10.1109/ICME46284.2020.9102829](https://doi.org/10.1109/ICME46284.2020.9102829).

- [16] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke, "Light field intrinsics with a deep encoder-decoder network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9145–9154, doi: [10.1109/CVPR.2018.00953](https://doi.org/10.1109/CVPR.2018.00953).
- [17] C. Shin, H.-G. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, "EPINET: A fully-convolutional neural network using epipolar geometry for depth from light field images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4748–4757.
- [18] L. Han, Z. Shi, S. Zheng, X. Huang, and M. Xu, "Light-field depth estimation using RNN and CRF," in *Proc. 7th Int. Conf. Image, Vis. Comput. (ICIVC)*, Jul. 2022, pp. 725–729, doi: [10.1109/ICIVC55077.2022.9886991](https://doi.org/10.1109/ICIVC55077.2022.9886991).
- [19] Y. J. Tsai, Y. L. Liu, M. Ouhyoung, and Y. Y. Chuang, "Attention-based view selection networks for light-field disparity estimation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020, vol. 34, no. 7, pp. 12095–12103.
- [20] J. Chen, S. Zhang, and Y. Lin, "Attention-based multi-level fusion network for light field depth estimation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 1009–1017.
- [21] H. Ma, H. Li, Z. Qian, S. Shi, and T. Mu, "VommaNet: An end-to-end network for disparity estimation from reflective and texture-less light field images," 2018, *arXiv:1811.07124*.
- [22] W. Zhou, X. Wei, Y. Yan, W. Wang, and L. Lin, "A hybrid learning of multimodal cues for light field depth estimation," *Digit. Signal Process.*, vol. 95, Dec. 2019, Art. no. 102585.
- [23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [24] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, Oct. 2014, pp. 1724–1734.
- [25] A. Graves and J. Schmidhuber, "Frame-wise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.
- [26] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, "ReNet: A recurrent neural network based alternative to convolutional networks," 2015, *arXiv:1505.00393*.
- [27] B. Shuai, Z. Zuo, B. Wang, and G. Wang, "Scene segmentation with DAG-recurrent neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1480–1493, Jun. 2018, doi: [10.1109/TPAMI.2017.2712691](https://doi.org/10.1109/TPAMI.2017.2712691).
- [28] A. C. S. Kumar, S. M. Bhandarkar, and M. Prasad, "DepthNet: A recurrent neural network architecture for monocular depth prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 396–3968, doi: [10.1109/CVPRW.2018.00066](https://doi.org/10.1109/CVPRW.2018.00066).
- [29] R. Kreuzig, M. Ochs, and R. Mester, "DistanceNet: Estimating traveled distance from monocular images using a recurrent convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1258–1266, doi: [10.1109/CVPRW.2019.00165](https://doi.org/10.1109/CVPRW.2019.00165).
- [30] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un)supervised learning of monocular video visual odometry and depth," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5550–5559, doi: [10.1109/CVPR.2019.00570](https://doi.org/10.1109/CVPR.2019.00570).
- [31] Y. Zhang and T. Chen, "Efficient inference for fully-connected CRFs with stationarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 582–589.
- [32] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1529–1537, doi: [10.1109/ICCV.2015.179](https://doi.org/10.1109/ICCV.2015.179).
- [33] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Monocular depth estimation using multi-scale continuous CRFs as sequential deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1426–1440, Jun. 2019, doi: [10.1109/TPAMI.2018.2839602](https://doi.org/10.1109/TPAMI.2018.2839602).
- [34] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Proc. Asian Conf. Comput. Vis. Cham, Switzerland: Springer*, 2016, pp. 19–34.
- [35] A. Mousnier, E. Vural, and C. Guillemot, "Partial light field tomographic reconstruction from a fixed-camera focal stack," 2015, *arXiv:1503.01903*.
- [36] M. Le Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1981–1993, Apr. 2018, doi: [10.1109/TIP.2018.2791864](https://doi.org/10.1109/TIP.2018.2791864).



LEI HAN received the B.S. and M.S. degrees in computer science and technology from the China University of Mining and Technology, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science and technology from Hohai University, China, in 2018.



SHENGNAN ZHENG received the B.S. degree from the Nanjing Institute of Technology, China, and the M.S. degree from Hohai University, in 2010, where she is currently pursuing the Ph.D. degree. She is currently with the Laboratory of Computer Engineering, Nanjing Institute of Technology. Her research interests involve artificial intelligence, machine learning, and computer vision.



ZHAN SHI received the B.S. degree from Zhengzhou University, China, and the Ph.D. and M.S. degrees from Hohai University. He is currently a Lecturer and a Teacher with the Institute of Computer Engineering, Nanjing Institute of Technology. His research interests involve artificial intelligence, machine learning, and software engineering.



MINGLIANG XIA received the bachelor's degree from the Harbin University of Science and Technology. He is currently a Professor with the Nanjing Institute of Technology, China. He has been engaged in engineering research on computer control for a long time and has undertaken a number of major projects in engineering development. His current research interests include computational imaging, artificial intelligence, and industrial control.

...