In this study, we propose a reconstruction and optimization neural network (RONN), a novel neural network for nonrigid structure from motion, which is completed by an unsupervised convolutional neural network. Compared with the traditional method for directly solving 3D structures, our model focuses on depth information that is lost owing to projection. This mathematical model is developed using a convolutional neural network with three modules for integration, reconstruction, and optimization, as well as two prior-free loss functions. The proposed RONN achieves competitive accuracy on several tested sequences and high visual quality of various real video sequences.

1. Introduction

Nonrigid structure from motion (NRSfM) targets the recovery of a nonrigid structure and camera matrix from given 2D point tracks in monocular views. Unlike its rigid counterpart [1], NRSfM is a highly ill-posed problem with several inherent ambiguities. Moreover, solving this problem requires additional constraints or priors. Many methods assume that the movement of the camera is slow and smooth [2–6]; however, this limits its applicability to real sequences. Another assumption is that the deformation of nonrigid instances can be represented using the weighted sum of basic deformations in the trajectory space [4] and shape space [7]. With these assumptions, the NRSfM problem is transformed into solving the basic deformation and its coefficients.

Inspired by these assumptions, many researchers have used neural networks to solve the sparse NRSfM problem [8, 9], which learns shape representations through unsupervised networks, while maintaining good generalization ability in the face of unseen data. However, their models are incapable of handling dense situations.

Dense NRSfM has achieved remarkable progress over the last several years [2, 10–13]. In 2020, Sidhu et al. [13] proposed the first dense neural NRSfM (N-NRSfM) approach with mean shape and demonstrated state-of-the-art performance on widely used datasets. However, when confronted with a long sequence or drastic changes, the mean shape is unreasonable. Additionally, it requires a considerable amount of time to obtain high-performance results.

In this study, we introduce a reconstruction and optimization neural network (RONN) and two improved loss functions for dense NRSfM. RONN mainly includes a depth reconstruction module and a camera optimization module, which reconstruct the depth information lost due to projection and optimize the camera matrix, respectively. Inspired by recent advances in NRSfM [8], the proposed improved loss function is combined with the minimum singular value ratio, and experiments show that it improves the original loss function to varying degrees.

The main contributions of our study are as follows.

1. We propose the first dense NRSfM network for reconstruction using depth information, namely, RONN. It is a convolutional neural network including reconstruction and optimization, which realizes the reconstruction of the 3D structure and the optimization of the camera matrix, respectively. Its specific structure will be introduced in Section 4.1.
2. Related Work

2.1. NRSfM. NRSfM is inherently ill-conditioned and requires additional constraints or priors to guarantee solution uniqueness. We are concerned with the following additional limitations:

(1) Bregler et al. [14] proposed a low rank, where the rank of the rigid 3D structure fixed is three. Dai et al. [7] rearranged the rows of S as $S^\#$ to obtain stronger low rank priors, demonstrating state-of-the-art performance on sparse datasets at the time. Ansari et al. [10] proposed scalable monocular surface reconstruction (SMAR) with an improved low rank. Its scalability enables the achievement of competitive accuracy on both sparse and dense data.

(2) Park et al. [15] proposed Procrustean regression, which is a regression problem based on Procrustes-aligned shapes. In [15], they proposed a novel regression framework for NRSfM, comprising Procrustes-aligned shape loss and low rank loss. The framework is versatile and can reconstruct a 3D structure under dense datasets. Additionally, Park et al. [16] proposed a novel framework for training neural networks with a Procrustean regression. Although the network structure is simple, it shows superior reconstruction performance compared to the state-of-the-art method. In [16], it was proven that Procrustes alignment could determine unique motions and eliminate the rigid motion components from reconstructed shapes.

2.2. Neural NRSfM. There have been studies on combining NRSfM with neural networks. Supervised neural networks require large amounts of training data; however, only a few datasets are currently available for quantitative evaluation of NRSfM methods. In contrast, unsupervised networks are easier to implement. C3DPO [17] and deep NRSfM [9] learn basis shapes from 2D observations without 3D supervision in sparse data sets. C3DPO [17], which was proposed by Facebook’s AI lab, uses a factorization network to replace the classical factorization step. Additionally, to ensure the effect of factorization, it collaborates with another canonicalization network to achieve a robust 3D reconstruction effect. This framework achieved high-performance reconstruction results for rigid and nonrigid datasets. Kong and Lucey [9] proposed a new a priori hypothesis, using multi-layer sparse coding to represent 3D nonrigid shapes, and designed an innovative encoder-decoder neural network to realize an unsupervised network for NRSfM. They extended the classic sparse coding algorithm, ISTA, to block sparse scenarios and provide state-of-the-art performance through the proposed network. However, sparse coding also limits its application to dense datasets. Sidhu et al. [13] introduced the first dense neural NRSfM approach, namely, N-NRSfM, and achieved competitive performance on widely used dense datasets. They used the mean shape to achieve reconstruction, but this became a limitation. Once the mean shape is determined, the reconstruction result is obtained. Therefore, when confronted with large-scale deformation, the reconstruction results are not as good as expected.

3. Mathematical Model

Consider a monocular camera for observing a nonrigid object with a set of P feature points. Let $S_f$ be the 3D shape matrix of the nonrigid object at the $f$th frame and $W_f$ be its 2D matrix according to an orthographic projection. Specifically,

$$S_f = \begin{bmatrix} x_f^1 & \cdots & x_f^P \\ y_f^1 & \cdots & y_f^P \\ z_f^1 & \cdots & z_f^P \end{bmatrix}, \quad S_f \in \mathbb{R}^{3 \times P},$$

(1)

$$W_f = \begin{bmatrix} u_f^1 & \cdots & u_f^P \\ v_f^1 & \cdots & v_f^P \end{bmatrix}, \quad W_f \in \mathbb{R}^{2 \times P}.$$  

(2)

$W_f$ and $S_f$ are related by the full rotation matrix $G_f$ as

$$W_f = \Pi G_f S_f,$$

(3)

where $W_f$ is already centralized; therefore, the camera matrix is reduced to pure rotation [14]. According to Formula (3), $W_f$ contains both the camera and 3D structure information. In this study, a reasonable network architecture was designed to separate the required information.

Formula (3) can be changed to the following form.

$$S_f = G_f^\top \begin{bmatrix} W_f \\ \varepsilon_f \end{bmatrix},$$

(4)

According to Formula (4), the full rotation matrix will have a certain effect on the reconstruction results; therefore, optimization of the full rotation matrix is necessary.
4. RONN Model

In this section, we introduce the structure and loss functions of the RONN.

4.1. Network Structure. As illustrated in Figure 1, inspired by the trajectory space, our RONN contains three modules: dimensional integration module \( f_D \), camera optimization module \( f_G \), and depth reconstruction module \( f_Z \). The dimensional integration module integrates the information contained in \( u \) and \( v \) in the \( W \) matrix, \( f_G \) represents the optimization module for the camera matrix, and \( f_Z \) represents the reconstruction module for the depth information \( z \).

The 2D matrix \( W \) first passes through a dimensional integration module, which consists of a convolutional layer with a kernel size of \( 2 \times 1 \) and a ReLU layer. The next two modules are \( f_G \), which has \( B \) (set to 1 by default) residual blocks and a linear layer after rearranging the shape, from which we obtain \( \Delta G_f \in \mathbb{R}^{3 \times 3} \), and \( f_Z \), which has \( Kx_z \) \( \in \mathbb{R}^{1 \times P} \). Specifically, the residual block contains two convolutional layers of kernel size \( 1 \times 1 \) and a ReLU layer.

Through this network, our reconstruction result can be expressed as

\[
S_f = \Delta G_f^T \cdot \Delta G_f \cdot \left[ \begin{array}{c} W_f \\ z_f \end{array} \right], \quad \Delta G_f = U V^T, \tag{5}
\]

where \( U \Sigma V^T = \Delta G_f \) is the SVD of the camera matrix.

In this study, the method for initializing \( G \) is the same as that in [1] on dense datasets and [7] on sparse datasets.

4.2. Loss Function. To solve the NRSfM problem, we propose minimizing the loss functions with the initial rotation matrix \( G \) as

\[
E = a E_{\text{temp}} + \beta E_{\text{PA}} + \gamma E_{\text{rank}}, \tag{6}
\]

where \( \{ E_{\text{temp}}, E_{\text{PA}}, E_{\text{rank}} \} \) encode the additional constraints.

The temporal smoothness term \( E_{\text{temp}} \) is used to constrain the similarity of the reconstruction results of adjacent frames as

\[
E_{\text{temp}} = \frac{1}{F-1} \sum_{f=1}^{F-1} \| \mu_f (S_{f+1} - S_f) \|_e, \tag{7}
\]

where \( \| \cdot \|_e \) denotes the Huber loss of the matrix. Weight \( \mu_f \) is discussed later.

The Procrustean alignment term \( E_{\text{PA}} \) can determine unique motions and eliminate the rigid motion components from reconstructed shapes [15]; the term can be expressed as follows.

\[
E_{\text{PA}} = \frac{1}{F} \sum_{f=1}^{F} \left\| \phi_f \cdot S_f - \bar{S}_f \right\|_e, \tag{8}
\]

where \( T = (1 - (1/P)11^T) \) is the translation matrix, centering the shape at the origin. Weight \( \phi_f \) is discussed later in this paper. This function is aimed at minimizing the error between the 3D shape of each frame and the reference shape to optimize the rotation matrix.

A rearranged shape matrix is expressed as

\[
S^\# = \begin{bmatrix}
X^1 \cdots X^p & Y^1 \cdots Y^p & Z^1 \cdots Z^p \\
\vdots & \ddots & \vdots \\
X_F^1 & \cdots & X_F^p & Y_F^1 & Y_F^p & Z_F^1 & \cdots & Z_F^p
\end{bmatrix},
\]

with an additional constraint \( \text{rank}(S^\#) < K \). In [10], they assumed that the mean 3D component was dominant in \( S^\# \) and could be removed in the temporal dimension. By combining both ideas, \( E_{\text{rank}} \) is defined as

Figure 1: Overview of the proposed RONN.

International Journal of Digital Multimedia Broadcasting

3
5. Experiment

In this section, the experimental results are described for several widely used benchmarks and real datasets. First, we introduce the datasets and experimental setups, then analyse and compare the proposed model with state-of-the-art dense and sparse datasets and, finally, use real data for experiments.

### 5.1. Datasets and Setups

#### 5.1.1. Datasets

Three dense benchmark datasets are used in the comparison of methods: synthetic faces (two sequences with 99 frames and two different camera trajectories denoted by Traj.A and Traj.B, with 28,887 points per frame) [2], expressions (384 frames with 997 points per frame) [18], and actor mocap (100 frames with 36,349 points per frame) [19].

#### 5.1.2. Evaluation Metrics

For algorithm performance indicators, the 3D error $e_{3D}$ defined as follows.

$$ e_{3D} = \frac{1}{F} \sum_{j=1}^{F} \left\| S_j^{GT} - S_j \right\|_F, $$(15)

where $\| \cdot \|_F$ denotes the Frobenius norm and $S_j^{GT}$ denotes the ground truth 3D structure at the $j^{th}$ frame.

#### 5.1.3. Training Details

The RONN was implemented in PyTorch [20]. We used the Adam optimizer with a learning rate of 0.0005 and trained for 2000 epochs. In the experiment, the weight was fixed at $\alpha = \beta = y = 1$.

### 5.2. Model Analysis

#### 5.2.1. Structure of RONN

The baseline was formed by removing the $f_G$ module in the RONN. These experiments show the advantages of $f_x$ and the necessity of $f_G$; it contains different combinations of loss functions on synthetic face sequences (Traj.A and Traj.B).

The advantages of $f_x$ are listed in Table 1. Because the $f_x$ network solves the depth information $z$, rather than the entire 3D structure, this allows the reconstruction to be achieved only with $E_{\text{temp}}$, although the error is relatively large.

The necessity of $f_G$ is shown in Table 2. When using the combination of $E_{PA}$ and $E_{\text{rank}}$, $e_{3D}$ is reduced by 31.72% for Traj.A and 16.88% for Traj.B. When using the combination of $E_{PA}$, $E_{\text{rank}}$, and $E_{\text{temp}}$, $e_{3D}$ is reduced by 33.83% for Traj.A and 22.00% for Traj.B.

### 5.3. Results

#### Table 1: $e_{3D}$ on baseline.

<table>
<thead>
<tr>
<th>Traj</th>
<th>$E_{\text{temp}}$</th>
<th>Baseline</th>
<th>RONN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$E_{PA}$</td>
<td>$E_{\text{rank}}$</td>
</tr>
<tr>
<td>Traj.A</td>
<td>0.0506</td>
<td>0.0457</td>
<td>0.0467</td>
</tr>
<tr>
<td>Traj.B</td>
<td>0.0670</td>
<td>0.0456</td>
<td>0.0459</td>
</tr>
</tbody>
</table>

#### Table 2: $e_{3D}$ on different loss function combinations.

<table>
<thead>
<tr>
<th>Traj</th>
<th>$E_{PA}$ $E_{\text{rank}}$</th>
<th>$E_{PA}$ $E_{\text{rank}}$ $E_{\text{temp}}$</th>
<th>$E_{PA}$ $E_{\text{rank}}$</th>
<th>$E_{PA}$ $E_{\text{rank}}$ $E_{\text{temp}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traj.A</td>
<td>0.0457</td>
<td>0.0467</td>
<td>0.0312</td>
<td>0.0309</td>
</tr>
<tr>
<td>Traj.B</td>
<td>0.0456</td>
<td>0.0459</td>
<td>0.0379</td>
<td>0.0359</td>
</tr>
</tbody>
</table>
Effectiveness of Improved Loss Functions. To understand the effectiveness of the improved loss functions, we conducted experiments with the following original loss functions.

\[
E_{\text{temp}} = \frac{1}{F-1} \sum_{f=1}^{F-1} \left\| S_{f+1} - S_f \right\|_e^c, \quad (16)
\]

\[
E_{\text{PA}} = \frac{1}{F} \sum_{f=1}^{F} \left\| S_f \cdot T - \bar{S} \right\|_e^c = \frac{1}{F} \sum_{f=1}^{F} S_f \cdot T. \quad (17)
\]

In Table 3, in the case of RONN without \( f_G \), compared with \( E_{\text{temp}}^* \), \( e_{3D} \) of the \( E_{\text{temp}} \) is reduced by 5.4% for Traj.A and 8.8% for Traj.B.

Because the combination of \( E_{\text{PA}} \) and \( E_{\text{rank}} \) must be used when using the \( f_G \) network, an experiment without \( f_G \) is added to show the performance of different improved functions. However, the improvement in \( E_{\text{PA}} \) did not affect the error before and after. However, with the addition of \( f_G \), the combination of \( E_{\text{PA}} \) and \( E_{\text{rank}} \) reduces the error by 12.11% on Traj.A and 5.01% on Traj.B, which also shows the necessity of optimizing the camera matrix.

5.3. Comparison of Methods

5.3.1. Synthetic Faces. \( e_{3D} \) for synthetic faces are listed in Table 4. Compared with jumping manifolds (JM) [21], Grassmannian manifold (GM) [11], SMSR [10], probabilistic point trajectory approach (PPTA) [22], consolidating monocular dynamic reconstruction (CMDR) [23, 24], variational approach (VA) [2], dense spatial-temporal approach (DSTA) [25], expectation-maximization finite element method (EM-FEM) [26], and N-NRSfM [13], the RONN achieves \( e_{3D} \) close to the best method on Traj.A and exhibits...
Table 6: $e_{AD}$ for the actor mocap dataset [19].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor mocap</td>
<td>0.054</td>
<td>0.0257</td>
<td>0.0226</td>
</tr>
</tbody>
</table>

Table 7: $e_{AD}$ for the sparse dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink</td>
<td>0.0227</td>
<td>0.0037</td>
<td>0.0266</td>
<td>0.0119</td>
<td>0.0147</td>
</tr>
<tr>
<td>Pickup</td>
<td>0.1791</td>
<td>0.0372</td>
<td>0.1731</td>
<td>0.0198</td>
<td>0.028</td>
</tr>
<tr>
<td>Yoga</td>
<td>0.1179</td>
<td>0.0140</td>
<td>0.1150</td>
<td>0.0129</td>
<td>0.0237</td>
</tr>
<tr>
<td>Stretch</td>
<td>0.1136</td>
<td>0.0156</td>
<td>0.1034</td>
<td>0.0144</td>
<td>0.0234</td>
</tr>
<tr>
<td>Dance</td>
<td>0.1877</td>
<td>0.1454</td>
<td>0.1864</td>
<td>0.1060</td>
<td>0.1404</td>
</tr>
<tr>
<td>Shark</td>
<td>0.1117</td>
<td>0.0135</td>
<td>0.2311</td>
<td>0.0551</td>
<td>0.0332</td>
</tr>
</tbody>
</table>

Figure 2: Visualizations of partial reconstruction result of real image sequences.
an average on $e_{3D}$ on Traj.B. Compared with Traj.A, the reconstruction accuracy of Traj.B is poor, as reflected by many methods.

5.3.2. Expressions. $e_{3D}$ for the expressions are presented in Table 5. Compared with the expectation-maximization linear dynamical system (EM-LDS) [3]; column space fitting, version 2 (CSF2) [27]; kernel shape trajectory approach (KSTA) [28]; global model with local interpretation (GMLI) [18]; and N-NRSfM [13], the RONN achieves $e_{3D} = 0.026$ on par with those of GMLI and N-NRSfM, which is currently the best method for this sequence. However, the number of iterations is significantly reduced (compared to 60,000 times in N-NRSfM).

5.3.3. Actor Mocap. $e_{3D}$ for the expressions are listed in Table 6. Compared with CMDR [23, 24] and SMSR [10], the RONN achieves $e_{3D} = 0.0226$, which is better than SMSR and CMDR.

5.3.4. Sparse Reconstruction. Except for the linear layer in $f_G$, the RONN is composed of a convolutional network, and each feature point shares parameters so that the network can handle datasets with different numbers of feature points. When facing classic sparse data, comprising six standard sequences, namely, drink, pickup, yoga, stretch, dance, and shark, the RONN could also realize reconstruction. The number of frames ($F$) and number of points ($P$), i.e., the $(F, P)$ set, for these datasets are $(1102, 41)$, $(357, 41)$, $(307, 41)$, $(370, 41)$, $(264, 75)$, and $(240, 91)$. As shown in Table 7, compared with the 3D reconstruction of dense data, in the sparse 3D reconstruction scene, the correlation between each point is relatively small owing to the few feature points. Compared with the classic sparse 3D reconstruction methods, the reconstruction results of the RONN are not as good as expected. However, even if the reconstruction error of RONN is not the best, it is also not the worst.

5.4. Experiments with Real Data. We also reconstructed several real image sequences, i.e., heart surgery [31], back [32], and real face [2] (see Figure 2). As for the real face, owing to the large amount of noise in matrix $W$, the final reconstruction result is not as smooth as expected. As for the back and heart surgery, the RONN achieved good visual reconstruction results.

6. Conclusion

This study proposes RONN and two improved loss functions. Our method can achieve reconstruction from 2D to 3D without supervision. One of the advantages of the RONN method is its scalability and consistent performance on datasets with different numbers of feature points.

As the first network to directly solve depth information to achieve reconstruction, RONN uses the depth reconstruction module $f_{3D}$, which can achieve 3D structure reconstruction with only temporal smoothness loss. Procrustean regression is used to optimize the camera matrix and improve performance and use msr to weight the above loss function to further improve the network’s ability to deal with complex deformation and experimentally demonstrate the improved loss function and the high performance of RONN.

Compared to the N-NRSfM approach, which is the first dense neural NRSfM, we do not need a mean shape and employ fewer training epochs. Compared to the classic sparse reconstruction method, the RONN shows better scalability. Transforming the input of the network from every frame to every point enables the network to better cope with the dense conditions.

Because of the direct use of $W$, the current limitation of the proposed method is its sensitivity to the 2D matrix $W$. As shown in Figure 2(c), the noise in $W$ is directly shown in the 3D structure, causing the result to be unsmooth. Since the original rotation matrix is required, the reconstructed structure accuracy will be affected by the original rotation matrix. Moreover, the RONN cannot handle data loss.

The N-NRSfM provides a new perspective on dense NRSfM, which we further improved, achieving results. In future research, we will consider complex situations, such as denoising and data loss.

Data Availability

The Python code data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Natural Science Foundation of Zhejiang Province (LZ20F020003, LY17F020034, LY17F020003, and LSZ19F010001) and the National Natural Science Foundation of China (61272311 and 61672466).

References


