Large parallax image matching based on Graph Matching

Guangmin Sun, Yu Weng* and Chong Shi

The Faculty of Information Technology, Beijing University of Technology, Beijing, China

*Corresponding author e-mail: wengyu@bjut.edu.cn

Abstract. In the security video surveillance system, image splicing technology faces the difficulty of achieving large parallax and small overlap area. This paper summarizes the traditional stitching technology and analyzes that the image matching algorithm is the key technology to solve the problem of large parallax stitching. Aiming at the problems that traditional matching methods with low matching accuracy cannot adapt to large parallax non-rigid transformatio, a model based on image matching is proposed in this paper for splicing tasks. The model which combines the similarity of the global structure and the local structure can not only divide the feature point on the structural level but also can pair with each other. At the same time, the framework can be trained end-to-end. The training samples composed of simulation data sets are adopted to train the designed model which is tested and verified by the test sample sets composed of three kinds of mixed data. Compared with PCA-GM, the model proposed in this paper adapts well to splicing tasks. The matching accuracy can achieve 93.1% and 91.9% on the simulation data set and the small parallax splicing data respectively, which can initially solve the image matching problem in the traditional image splicing scene. However, a further research aiming at the problem of the matching accuracy on the large parallax data set can only achieve 74.5% will be needed.

1. Introduction

With the demand of security in various fields, the installation of video surveillance systems has become more and more essential. However, due to the limitations of the area monitored by a single camera, it is necessary to monitor image scenes under different viewing angles at the same time to obtain more extensive information. By using video splicing technology, videos collected by multiple cameras placed under different viewing angles can be spliced in multiple channel, which can form a richer video image. However, due to the constraints of cost and layout, the overlapping area between surveillance cameras is small and the same scene is presented with a large parallax.

In 2013, TJ CHIN proposed APAP, which divided the image into grids and aligns them with a homography matrix [1]. It was a milestone for the problem of large parallax stitching. Then, some algorithms such as AANAP [2], GSP [3], SEAGULL [4] had all made improvements. However, the overlapping area basically contains the entire image when the parallax is large in the above method.

In the process of image stitching, the stitching task of large parallax and small overlap area encounters difficulties in the module of image matching. Common algorithms of indirect matching need to combine the setp of feature extraction. However, such methods require operations of post-processing.
and cannot perform non-rigid transformation. The algorithm of direct matching cannot be solved in polynomial time and it can be also affected by noise and outliers.

Based on the deep graph matching in deep learning, the splicing problem of large parallax was studied in this paper. Under the assumption that the extraction of feature point is completed, the matching problem can be converted into a problem of corresponding the two point sets. Deep graph matching based on graph neural network (GNN) can use the structure of graph to build a model of image feature points, which can effectively solve the problem of non-rigid transformation. In addition, the structure of graph is naturally suitable for the study of the problems of corresponding the point sets.

2. Materials and Methods

2.1. Data Collection and Analysis

Since there are few public and available data sets for splicing tasks, establishing a simulation data set for training is needed in this paper. The VOC2011 image feature point detection data set was adopted to cut the original image twice in different scales [5]. On the basis of preserving the integrity of the label, one image can generate two images to be stitched. After that, in order to fit the data feature of the actual stitching task, every image was randomly processed with the operation of scale, rotation, lighting, and noise. Similarly, the coordinate values of keypoints were blurred by random affine transformation and another random noise of $N(0, \sigma^2)$ added. Finally, 7146 pairs of stitched images were obtained and every image carried keypoint coordinate label information.

Two identical surveillance cameras whose resolution is 1080p were adopted to collect spliced test data in large parallax. In addition, it was important to increase the distribution of data in different scenes and lighting. The samples must meet the following conditions: First, there must be a certain overlap area between the samples; Second, the camera must ensure that the image sequence must be acquired in a scene.

According to the different parallax can be divided into two types: horizontal parallax 0-75 degrees, vertical parallax 0-40 degrees. The data contained 24 pairs of samples whose horizontal parallax was lower than 35 degrees and whose vertical parallax was lower than 20 degrees can be regarded as the small parallax data set. The remaining data contained 24 pairs of samples can be regarded as the large parallax data set. Meanwhile, the coordinates of the keypoints were labeled manually.

Finally, the simulation data set was divided into training set and testing set according to a ratio of 3:1. In addition, the collected data sets with large and small parallax are adopted as the testing set. The Training Size and Testing Size are shown in Table 1. The data sets are shown in Figure 2 and Figure 3, which are respectively stand for the simulation data set, the small parallax data set and the large parallax data set.

| data set name          | parallax | Training Size | Testing Size |
|------------------------|----------|---------------|--------------|
| simulation data set    | Non      | 5360          | 1789         |
| small parallax data set| small    | 24            |              |
| large parallax data set| large    | 48            |              |

2.2. Propose Approach

Based on the PCA-GM (permutation loss and cross graph affinity) model [6], an suitable deep graph matching model was set up for stitching tasks. The model proposed in this paper consisted of feature extraction module, graph embedding module and relationship extraction module. In the feature extraction module, CNN was adopted to extract the image features of keypoints as the node features which input into the graph network. Then, feature aggregation was performed by the graph embedding module composed of intra-graph matching (Intra-GM) and cross-graph matching (Cross-GM). Meanwhile, the affinity matrix of cross-graph was calculated by softmax and passed back to the Intar-
GM. Finally, the matching relationship matrix was predicted by the activation function of sigmoid. The input of the network was the pixel and the keypoint coordinates of the two images. The output was the cross-graph affinity matrix adopted to predict the correspondence relationship of node-to-node. The entire structure is shown in Figure 1.

![Figure 1. Overview of our approach.](image)

Compared with PCA-GM, the model proposed in this paper simplified the graph embedding module, which made and the combination of modules have been greatly changed. In addition, the cross-graph affinity matrix was added to the graph structure for iteration. For the asymmetric matching problem in splicing, the activation function of softmax was adopted to replace the sinkhorn algorithm.

2.2.1. Feature Extraction. When the keypoints coordinate set \( P = \{p_i\} \) was known, the features of the keypoints were constructed by interpolation on CNN’s feature map.

\[
\text{f}(0) = \text{interp}(P, \text{CNN}(I))
\]

where \( \text{CNN}(I) \) performed CNN on image I and output a feature tensor \( X \). \( \text{interp}(P, X) \) interpolates on the point \( P \) of the tensor \( X \) via bilinear interpolation. Two input images can share the same CNN structure and weights. Finally, the visual feature set \( F = \{f_i; i \in N\} \) was obtained.

Although CNN can implicitly encode the position information, this information is provided by zero-padding. The representation of the location information was enhanced through a multimodal approach. In the image \( I_i \), the geometric feature \( p_i \) and the visual feature \( (F_{ij}) \) of each keypoint were concatenated to obtain the node feature \( N_i = (p_i, (F_{ij})) \). After obtaining a feature vector for every keypoint, the graph structure was constructed by Delaunay triangulation.

2.2.2. Intar-graph Matching. In Intar-GM, graph convolutional network (GCN) was adopted as the method of graph embedding module. The function of GCN was to effectively aggregate the nodes and the adjacent nodes \[7\]. Then introducing convolution calculations to enhance the learning efficiency of the network. Intar-GM transferred feature information between adjacent nodes through GCN and embedding the information of the graph structure in the feature vector of the node to reflect the feature of the graph structure.

The delivery method of Intar-GM is shown as follows:

\[
m_{st}^{(k)} = \frac{1}{|\{(i,j)\in E_s\}|} \sum_{(i,j) \in E_s} f_{\text{ReLU}}(n_{st}^{(k)})
\]

\[
n_{st}^{(k+1)} = f_{\text{update-intra}}(m_{st}^{(k)}, n_{st}^{(k)})
\]

\( f_{\text{ReLU}} \) standed for the function of ReLU activation, and \( f_{\text{update-intra}} \) standed for summation function. \( n_{st}^{(k)} \) standed for the i-th node vector of the k-th layer in the graph s, \( m_{st}^{(k)} \) standed for the i-th message passing vector of the k-th layer and \( E_s \) standed for the set of the graph s’s edges.
In order to avoid the bias due to the different number of neighbors owned by different nodes, Eq. (2) normalized the aggregation features of the adjacent nodes according to the total number of adjacent nodes, which is a common practice of GCN. Eq. (3) adopted accumulated information to update the state of node $i$. 

### 2.2.3. Cross-graph Affinity Matrix.

For the graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the network obtained a feature vector $\{n_{1i}^{(k)}, n_{2j}^{(k)}\}$ of $k$-th layer after Intar-GM, $i \in V_1, j \in V_2$. An affinity matrix $M$ was constructed by calculating the affinity of any two vectors between the two graphs:

$$
M_{ij}^{(0)} = \exp\left(\frac{n_{1i}^{(k)}n_{2j}^{(k)}}{\tau}\right), i \in V_1, j \in V_2
$$

$M_{ij}^{(0)}$ stood for the similarity between node $i$ in the first graph and node $j$ in the second graph, which meant the similarity score between the two graphs. The exponential function could ensure the output of a positive number. Considering that the feature vector had $m$ dimensions, which meant $\forall i \in V_1, j \in V_2, n_{1i}, n_{2j} \in R^{m \times 1}$. $A \in R^{m \times m}$ contained learnable weights of this affinity function. $\tau$ stood for a hyperparameter that adjusted the discriminative ability of Eq. (4). For $\tau > 0$, the Eq. (4) became more discriminative when $\tau \to 0^+$.  

After calculating the affinity matrix $M$, the function of softmax activation was adopted to solve the relationship matrix in the row dimension of the matrix. When the temperature parameter of softmax tended to 0, softmax would become a one-hot vector:

$$
S = Softmax(M^{(0)})
$$

The matching relationship matrix $S$ would participate in the updation of Cross-GM.

### 2.2.4. Cross-graph Matching.

Cross-GM needed to transfer features between two graph structures which were aggregated from nodes with similar features in other graphs. Cross-GM which was similar to Intar-GM adopted GCN for message transmission.

The transfer method of Cross-GM is shown as follows:

$$
\begin{align*}
m_{1i}^{(k)} &= \sum_{j \in V_2} S_{ij} f_{map}(n_{2j}^{(k-1)}) \\
h_{1i}^{(k)} &= f_{update-cross}(m_{1i}^{(k)}, n_{1i}^{(k)})
\end{align*}
$$

Eq. (6) aggregated features from other graphs, $f_{map}$ was adopted as identity mapping, $S_{ij}$ standed for the matching relationship matrix. Eq. (7) was adopted to accumulate information to update the state of node $i$ in $G_1$, $f_{update-cross}$ standed for the concatenation of two tensors of input feature, which was followed by a fully-connected layer.

The calculation steps of Cross-GM are shown as follows:

First, input the feature vector $\{n_{1i}^{(k)}, n_{2j}^{(k)}\}, i \in V_1, j \in V_2$ after Intar-GM. Subsequently, the similarity of any two vectors between the two graphs could be calculated by Eq. (5). Then an affinity matrix $M$ was constructed.

The second step was to use the function of softmax activation on the affinity matrix $M$ to extract a cross-graph matching relationship matrix $S$. It should be noted that $S$ standed for the predicted correspondence relationship from $G_2$ to $G_1$ while $S^\top$ standed for the relationship from $G_1$ to $G_2$.

Finally, this cross-graph matching relationship matrix $S$ was adopted as the update weight between the two graphs. The more similar node pairs were in the feature $\{n_{1i}^{(k)}, n_{2j}^{(k)}\}, i \in V_1, j \in V_2$, the higher the weight would be updated when updating across graph.

### 2.2.5. Relation Extraction.

After the affinity matrix $M$ was calculated in the last layer, the function of sigmoid activation could be adopted to solve every row of $M$:

$$
S^{pred} = Sigmoid(M^{(k)})
$$
After setting the threshold $t$, all the element set $\{S_{ij}^{\text{pred}}\}$ of $S^{\text{pred}}$ which was greater than $t$ could be extracted. Then, the candidate node relationship set $\{(v_i, v_j)\}$ could be found through the subscript.

If there was a relationship which was greater than 1 set for the vertex $v_i$, the vertex corresponding to the element with the largest probability value in the set $\{S_{ij}^{\text{pred}}\}$ would be selected, which represented a one-to-one correspondence relationship. Finally, the predicted matching relationship matrix $S^{\text{pred}}$.

### 2.2.6. Loss Function.

The node-to-node correspondence in ground truth was adopted as the supervised information for the end-to-end training. In addition, permutation loss based on cross entropy was adopted as well.

$$
-\sum_{i \in V_1, j \in V_2} \log S_{ij}^{\text{gt}} \log S_{ij}^{\text{pred}} + \log(1 - S_{ij}^{\text{pred}}))
$$

### 3. Result

In the case where the coordinates of the feature point were known information, the predicted matching accuracy of the network could be solved by the matching relationship matrix $S^{\text{pred}}$ and the ground truth permutation matrix $S^{\text{gt}}$:

$$
\text{acc} = \frac{\sum \text{AND}(S_{ij}^{\text{pred}}, S_{ij}^{\text{gt}})}{N}
$$

Where AND standed for a logical function. The matching accuracy could be calculated by dividing the number of correctly matched keypoint pairs by the total number of keypoint pairs.

Comparative experiments were conducted with PCA-GM based on the same test conditions. A simulation data set was adopted in the experiment. Among the data set, 5360 pairs of images were adopted as the training set, 10720 images were in total; the remaining 1786 pairs of images were adopted as the verification set. In addition, the small and large parallax data sets were adopted as test sets. The specific test results are shown in Table 2.

The experimental results showed that under the same test set sample conditions, the model proposed in this paper was more suitable for image stitching tasks than PCA-GM. The model proposed in this paper possessed a matching accuracy of 91.9% on the small parallax test set, which indicating that the model could achieve better results in traditional stitching tasks. The testing results are shown in Figure 2.

For the large parallax test set, the matching accuracy could reach 74.5%. For large parallax images whose camera's horizontal deviation exceeds 75 degrees and whose vertical deviation exceeds 20 degrees, there will be misconnection problems. The model also needs to be improved for the problem of large parallax image registration. The testing results are shown in Figure 3.

![Figure 2. Matching of small parallax data.](image-url)
4. Conclusion
Aiming at the problem of low matching accuracy of traditional registration methods in large parallax image stitching, a new deep graph matching model based on PCA-GM is proposed in this paper. The network is composed of feature extraction module, graph embedding module and relationship extraction module. The results showed that under the same test conditions, the model proposed in this paper not only possesses a higher matching accuracy than PCA-GM but also is well adapted to the stitching task in the stitching scene. However, the matching accuracy of images with large parallax is still needed to be improved.

References
[1] Zaragoza J, Chin T J, Brown M S, et al. As-projective-as-possible image stitching with moving DLT[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2013: 2339-2346.
[2] Lin C C, Pankanti S U, Natesan Ramamurthy K, et al. Adaptive as-natural-as-possible image stitching[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1155-1163.
[3] Chen Y S, Chuang Y Y. Natural image stitching with the global similarity prior[C]//European conference on computer vision. Springer, Cham, 2016: 186-201.
[4] Lin K, Jiang N, Cheong L F, et al. Seagull: Seam-guided local alignment for parallax-tolerant image stitching[C]//European conference on computer vision. Springer, Cham, 2016: 370-385.
[5] Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge[J]. International journal of computer vision, 2010, 88(2): 303-338.
[6] Wang R, Yan J, Yang X. Learning combinatorial embedding networks for deep graph matching[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 3056-3065.
[7] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.