Directly Obtaining Matching Points without Keypoints for Image Stitching

Yujie Huang, Minge Jing, Yibo Fan, Xiaoyong Xue, Xiaoyang Zeng

State Key Laboratory of ASIC and System, Fudan University, China

2020 IEEE International Symposium on Circuits and Systems

Virtual, October 10-21 2020

Outline

- Background
- Our Algorithm
- Experimental Results
- Conclusion

Background

What is image stitching?

How?

Extract Keypoints

Match Keypoints

Map

Blend

Background

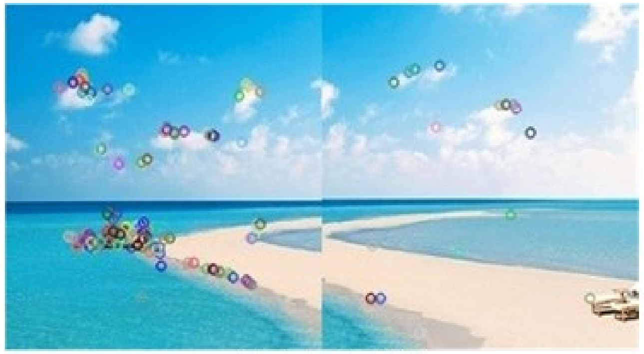
Main Challenges in Image Stitching

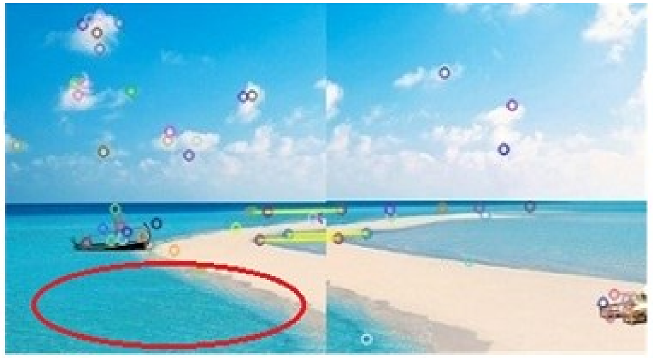
- Large Parallax
- Detectable Features are Not Obvious (DFNO)
- Large Differences Between Two Images
- Small Overlapping Area

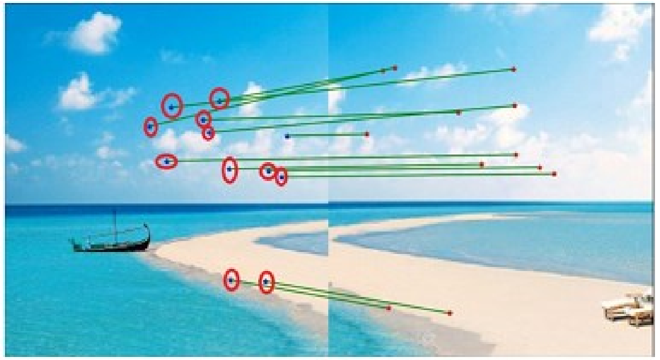
Large Parallax

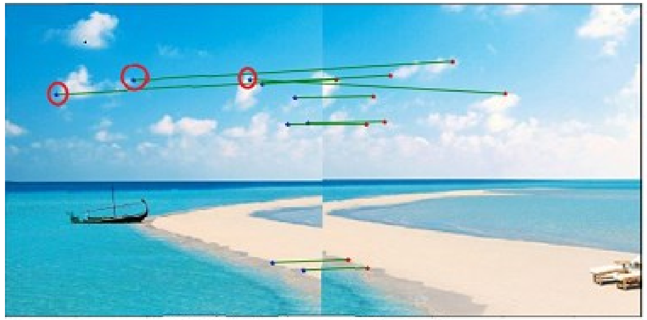
Background

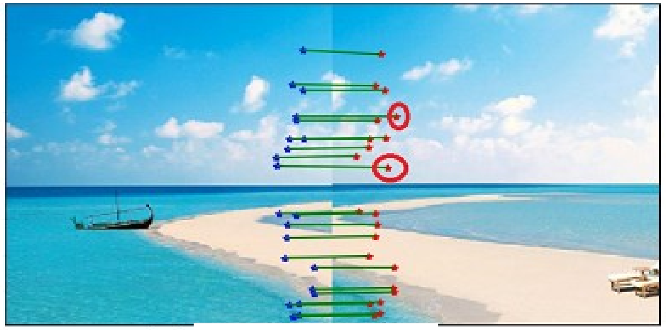
Detectable Features are Not Obvious


(a)ORB


(b)SIFT

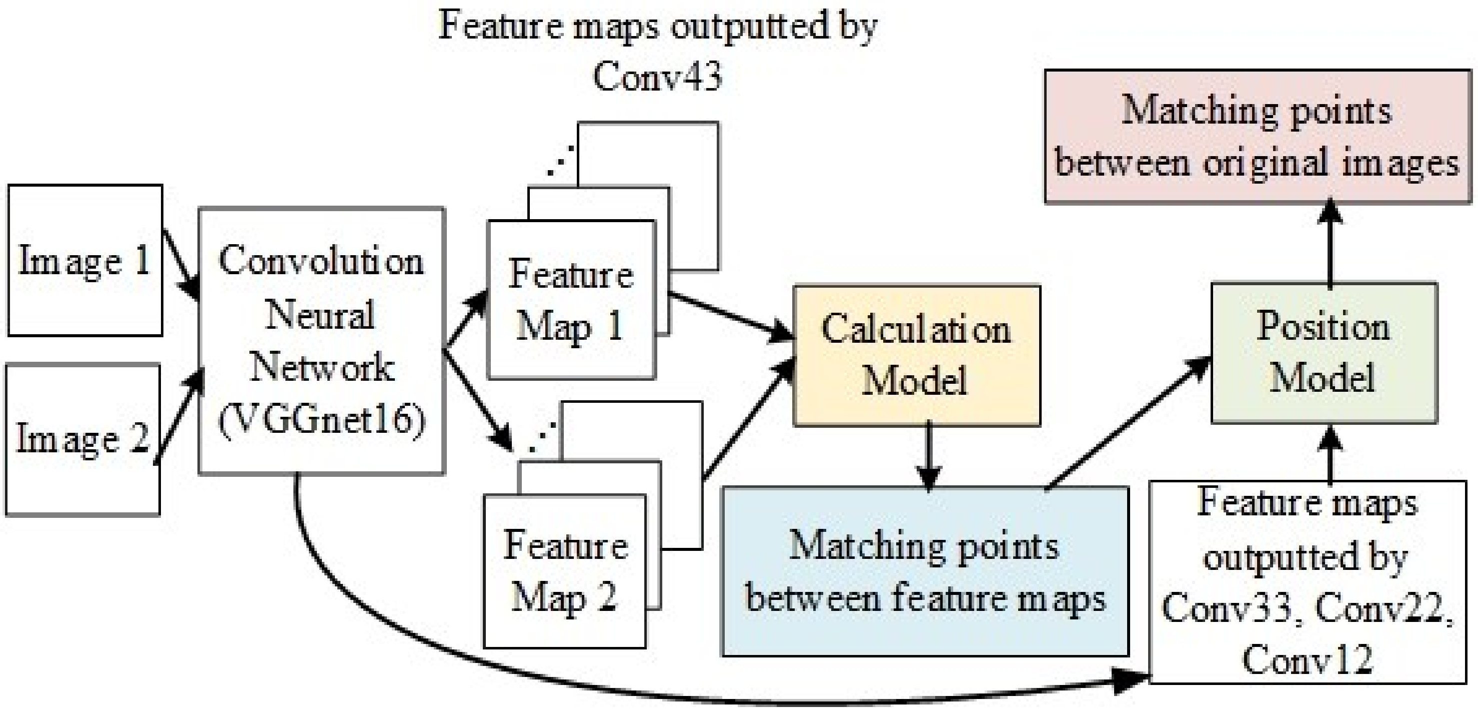

(c)LIFT


(d)SuperPoint


(e) Ours

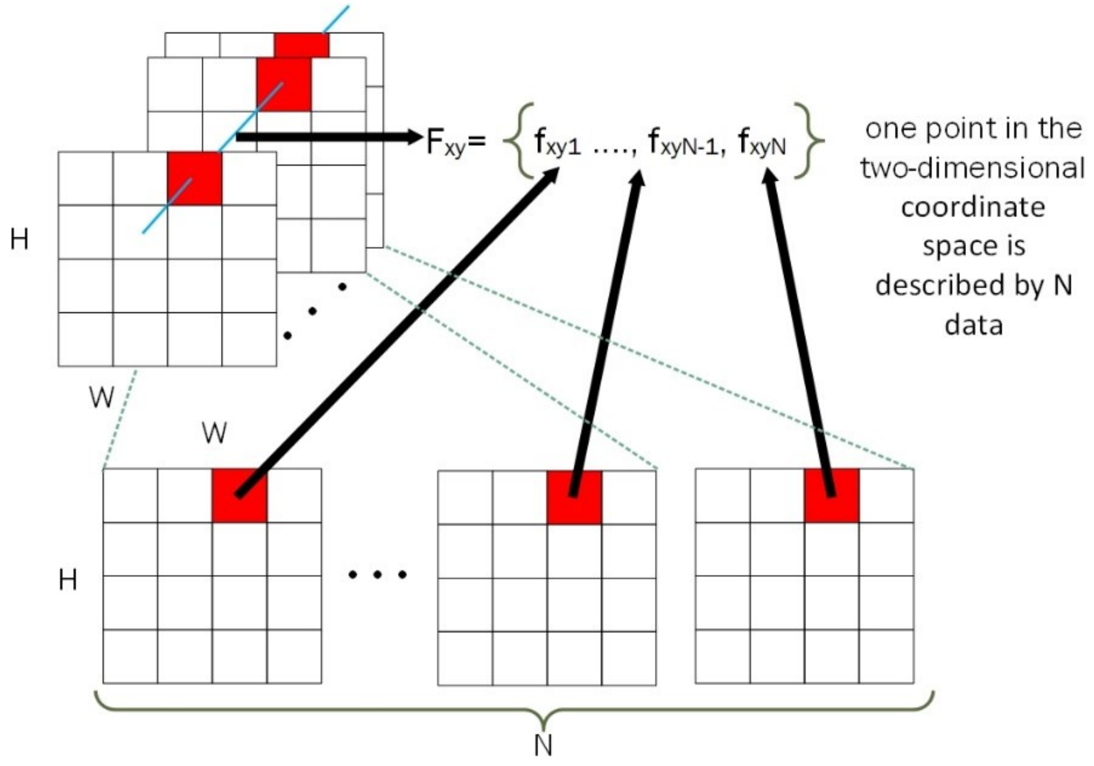

(f)Stitched image

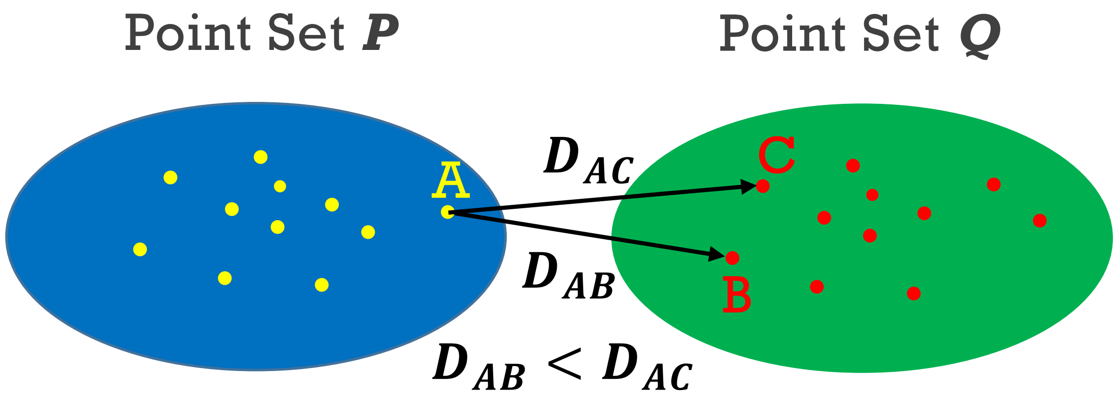
Our Algorithm



1. extract the feature maps of two images by a pre-trained VGGNet16.
2. obtain the matching points between the feature maps of two images based on the calculation model.
3. map the matching points to the original images based on the position model.

Our Algorithm-Calculation Model





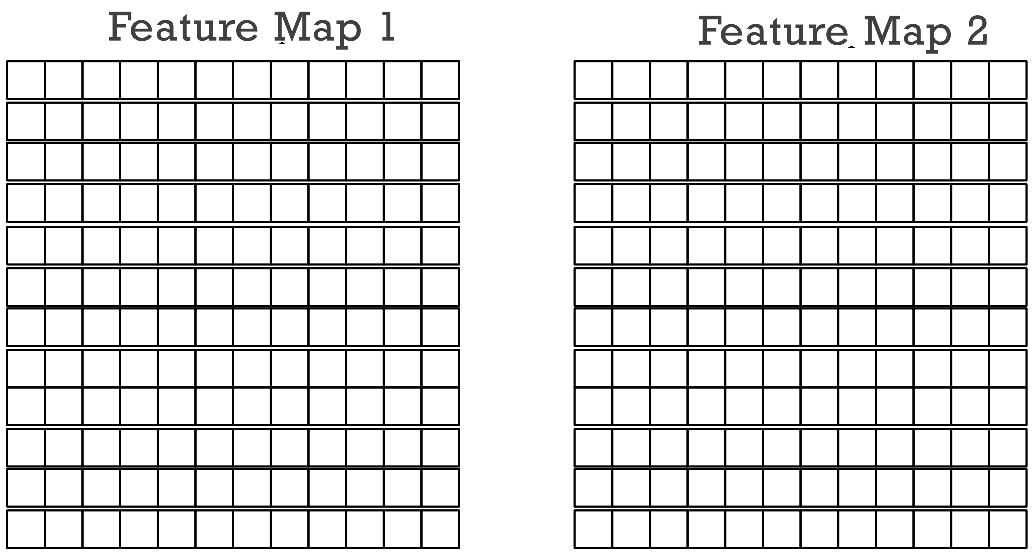
$$F_{xy} = \{f_{xy1}, f_{xy2}, \ldots, f_{xyN}\}$$

$$D_{ab} = \sum_{i=1}^{N} |f_{x1y1i} - f_{x2y2i}|$$

$$\frac{D_{AB}}{D_{AC}} \leq T$$

Our Algorithm-Obtain Matching Points

Brute-force Matching is Time-Consuming

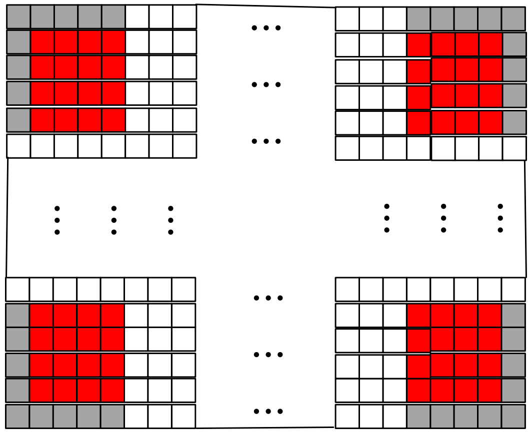


Divide the matching process into two steps:

1) Pre-locate

2) Fine-locate

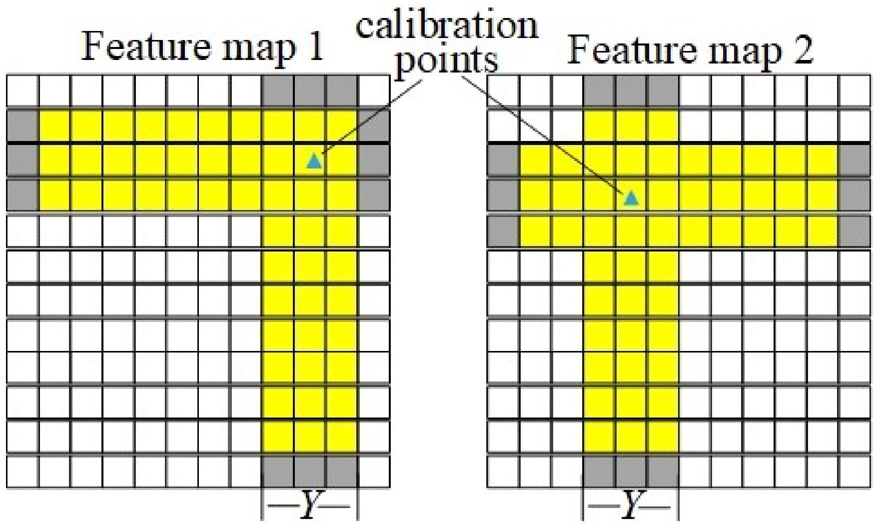
1) Pre-locate



2) Fine-locate

Y is 1/32 of the long dimension of the image

For example, Y is 13 for image with size 420*400

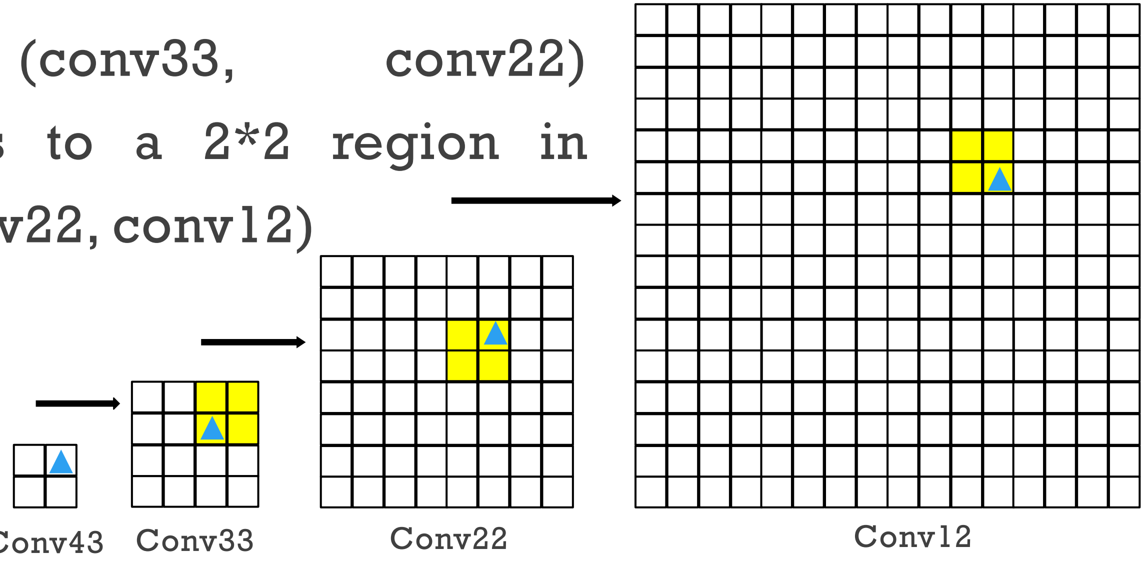


Our Algorithm-Position Model

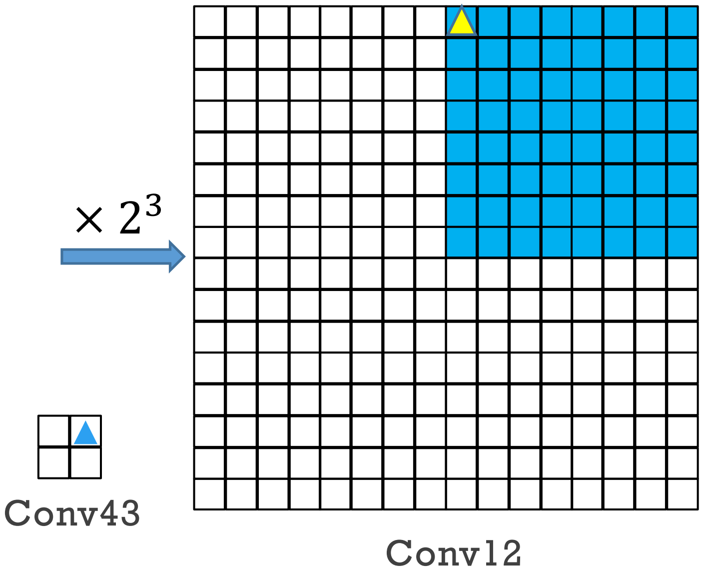
Matching Points between Feature Maps (Conv43)

conv43 (conv33, conv22) corresponds to a 2*2 region in conv33 (conv22, conv12)

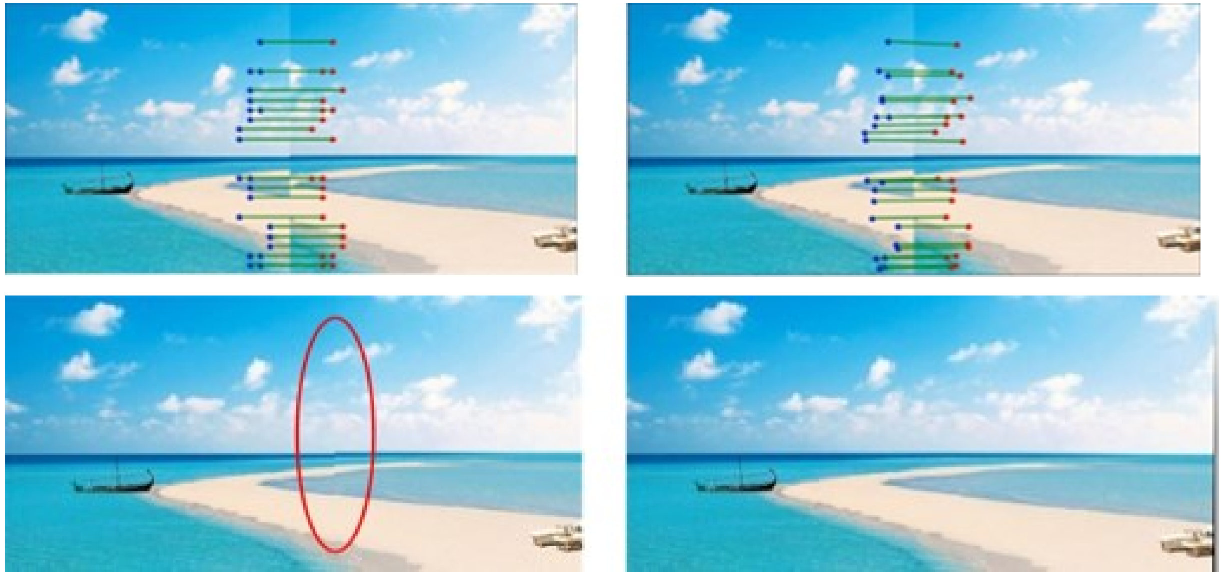
2*2 maxpooling operation



Map them to the original image



$\times 2^3$

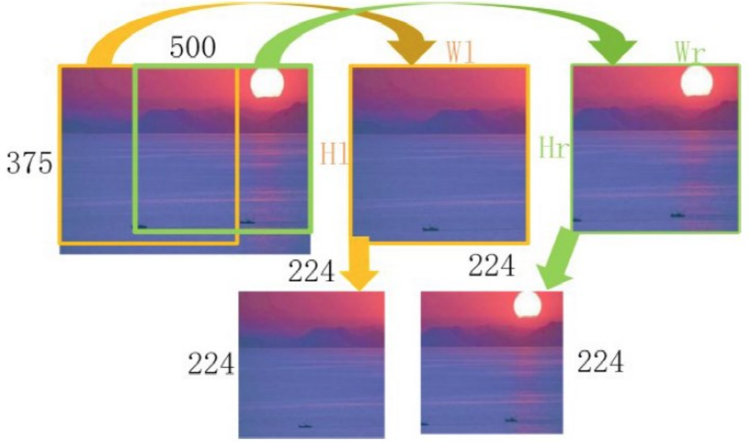


Position Model

Experimental Results-Create DataSet

Detectable features are not obvious

50 images (landscape class in Baidu image database)

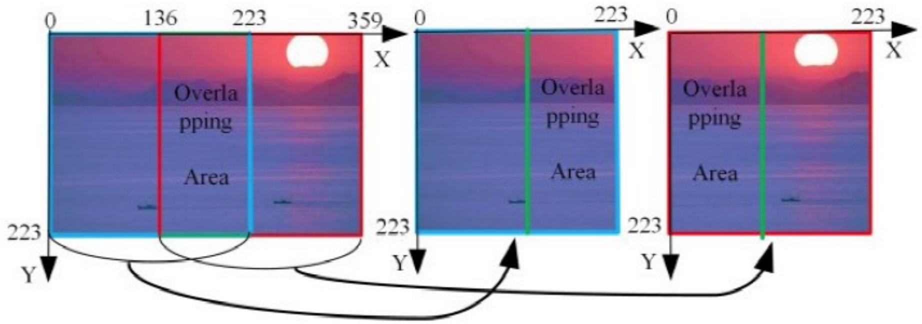


Different Scale

$$\begin{cases} \frac{W_r}{224} \cdot x_r = \frac{W_l}{224} \cdot x_l - (500 - W_r) \\ \frac{H_r}{224} \cdot y_r = \frac{H_l}{224} \cdot y_l \end{cases}$$

Simple Split

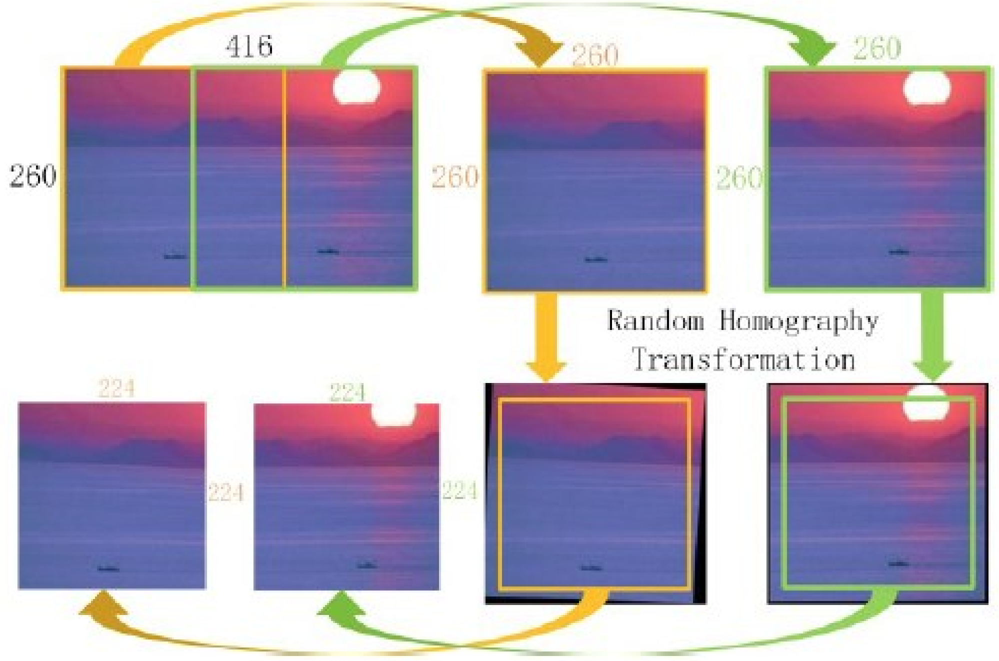
$$\begin{cases} x_r = x_l - 136 \\ y_r = y_l \end{cases}$$



Homography

$$\begin{cases} C_l = M'_l \times (x_l + 18, y_l + 18, 1)^T \\ C_r = M'_r \times (x_r + 18, y_r + 18, 1)^T \end{cases}$$

$$\begin{cases} \frac{C_r[0]}{C_r[2]} = \frac{C_l[0]}{C_l[2]} - 156 \\ \frac{C_r[1]}{C_r[2]} = \frac{C_l[1]}{C_l[2]} \end{cases}$$



Normal Case

50 images from the ILSVRC2012

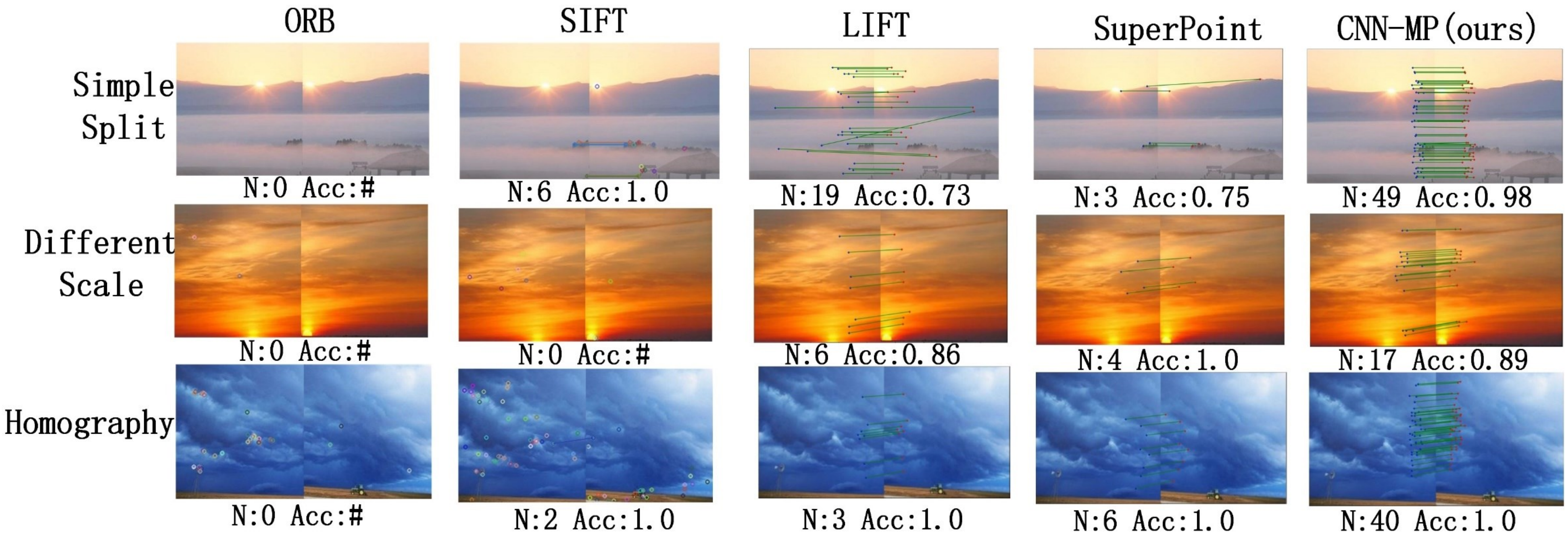
Experimental Results

TABLE I. COMPARISON RESULT IN THE DFNO CHALLENGE

Algorithm	Simple Split		Different Scale		Homography	
	N	*Acc*	*N*	*Acc*	*N*	*Acc*
ORB	3.5	**1.0**	3.9	0.9	0.26	0.565
SIFT	17.1	0.966	10.9	0.957	3.5	0.673
LIFT	25.1	0.83	14.6	**0.962**	5.5	0.849
SuperPoint	10	0.912	10.3	0.87	3.2	0.675
CNN-MP	**54.8**	0.993	**25.7**	0.885	**22.70**	**0.895**

TABLE II. COMPARISON RESULT IN THE NORMAL CASE

Algorithm	ORB	SIFT	LIFT	SuperPoint	CNN-MP
N	3.08	44.02	8.08	9.92	**51.02**
Acc	0.69	0.811	**0.906**	0.718	0.896



Experimental Results

Conclusion

- The challenge where detectable features are not obvious
- Propose a method called CNN-MP
- Break the conventional image stitching steps
- Calculation Model
- Pre-locate and Fine-locate
- Position Model
- More accurate matching points in both cases where detectable features are obvious and not obvious

Tank You!

Q&A