Object Detection on Panoramic Images Based on Deep Learning

Fucheng Deng, Xiaorui Zhu*, Jiamin Ren
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
e-mail: xiaoruizhu@hit.edu.cn

***Abstract*—Panoramic image can be widely used in many applications, such as virtual reality, visual surveillance and autonomous vehicle, because of its large field of view. However, the inherent distortion for panorama causes object detection to be a challenging task. This paper focuses on the multi-class objects detection in panoramic images using deep learning method. The proposed system uses three fisheye cameras to efficiently create panoramas and build a large dataset. A region based convolutional neutral network (R-CNN) is implemented to train and test on an indoor panoramic image dataset. Experiments show great improvement performance on ten categories of distorted indoor objects with a mean average precision of 68.7%.**

Keywords-object detection; panoramic image; deep learning; R-CNN

I. INTRODUCTION

Panoramic image has enjoyed popularity in many applications, such as virtual reality, visual surveillance, and autonomous vehicle [1]-[3]. Particularly, we are interested in the virtual interaction for the real-estate industry aiming to make it more convenient and enjoyable for the people to view the house or apartment without onsite inspection. Basically, it's important to collect the panoramic image of the rooms and automatically recognize and detect the objects in them. A full panoramic image can be obtained using panoramic acquisition device consist of multi fisheye cameras. For such device, multiple fisheye images are simultaneously captured to construct panorama through image stitching. However, fisheye lens with large field of view produce severe distortion and cause deformed objects in the constructed panoramic image. The distortions of objects vary with distance and viewpoint and show randomness to some extent (e.g., Fig. 1). Thus, objects detection in such panoramic images is challenging. Only a few literatures are reported to detect specific kinds of objects such as pedestrians and vehicles in fisheye image [4], [5]. To our knowledge, this is the first time to report multi-class objects detection in panoramic images.

Traditional object detection algorithms are usually based on hand-crafted features, such as Scale-Invariant Feature Transform (SIFT) [6], Local Binary Pattern (LBP) [7], Histogram of Oriented Gradient (HOG) [8] and Aggregated Channel Feature (ACF) [9]. Features are extracted in the image window and processed with specific learning algorithm. The most widely used algorithms are based on SVM, AdaBoost and Random Forest [10], [11]. The object with unknown size and aspect ratio could be anywhere in the image. Sliding-window adopted by many traditional detection methods has to be applied to scan the whole image in order to localize the object. Unfortunately, the sliding-window based approach is not efficient for the detection of multiple deformed objects in panoramic image because the scales and aspect ratios of deformed objects differ greatly. It is hard to design the size of sliding-window. An alternative method is to design a group of sliding-windows with different sizes. But this is at cost of computation and the performance is still not satisfying.

Figure 1. Strongly distorted object in a panoramic image

In recent years, deep learning has achieved unprecedented success in image classification and detection. Especially, convolutional neural network (CNN) has played an important role in the development of deep learning and been the most widely used deep network in modern applications [12]. Compared with traditional features, CNN feature is more abstract and representative. Furthermore, CNN feature is more robust to geometric transformation, distortion, and illumination. In this paper, region based CNN (R-CNN) [13], [14] is used for multiple objects detection in panoramic image. This method incorporates the idea of region proposal to avoid the problem of sliding-window and thus is adaptive to our application.

The main contributions of this paper can be briefly described in twofold. Firstly, deep learning approach has been extended for multi-class objects detection in the context of strong distortions in panoramic images. Secondly, a fast method for panorama construction is proposed and a panoramic image dataset for indoor environment has also been created for training and testing purposes. The remainder

978-1-5090-6086-3/17/$31.00 ©2017 IEEE

of this paper is organized as follows. The construction of panorama from fisheye images is presented in section II. The object detection process is discussed in section III. Section IV presents the experimental results. Finally, conclusions are given in section V.

II. PANORAMIC IMAGE GENERATION

A. Overview of the Image Acquisition

A 360 degree panoramic camera system has been designed using three annular horizontally placed fisheye cameras (Fig. 2). The original image acquisition from three cameras can be performed simultaneously. It avoids shooting a sequence of images at a certain rotation angle with a single camera. The image processing, including distortion correction and image mosaic, runs in both CPU and GPU processors. And thus it can be used to make a real-time panoramic video system. Furthermore, an interface unit has been incorporated to upload and share the panorama data in a remote server. Therefore, it is very convenient to use such device to collect a large number of images since big data is required for deep learning.

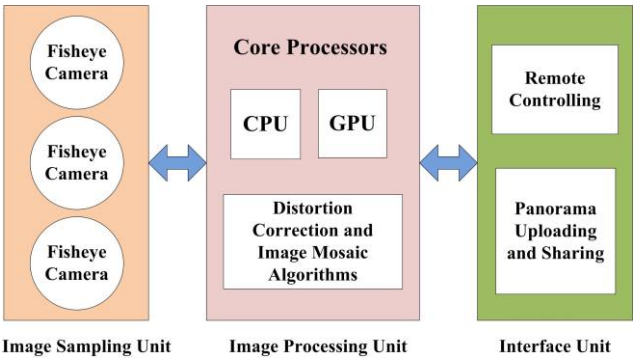


Figure 2. Overview of the panoramic image generation system

B. Distortion Correction

There are many classical distortion correction methods such as spherical projection model, polynomial transformation model and latitude-longitude transformation model [3]. For the purpose of object detection, the correction accuracy is not the key issue. Thus, a simple and fast approach based on longitude-latitude projection is proposed in this paper. The idea of longitude-latitude projection is to project the distorted fisheye images into square shape in a way of longitude and latitude as general photos. There are mainly four steps. Firstly, the pixel coordinate in the corrected longitude-latitude image (i, j) is transformed into latitude-longitude coordinate (θ', φ') according to

$$\theta' = \pi j / h, \varphi' = 2\pi i / w \tag{1}$$

where *w* and *h* are the width and height of image respectively.

Secondly, the mapping relationship between the real spatial coordinate (x, y, z) and the latitude-longitude coordinate (θ', φ') in a unit spherical coordinate system is

$$x = \sin\theta' \cos\varphi', y = \sin\theta' \sin\varphi', z = \cos\theta' \tag{2}$$

Thirdly, a new spherical coordinate (θ, φ) can be obtained through

$$\theta = \cos^{-1}(y), \varphi = \tan^{-1}(z / x) \tag{3}$$

The fisheye image could be considered as a projection from sphere to the plane of *xoz* in a way of concentric circles. The relationship between the polar coordinate of fisheye image plane (r, ϕ) and the spherical coordinate (θ, φ) could be expressed as

$$r = 2\theta / F, \phi = \varphi \tag{4}$$

where the *F* is the field of view for the fisheye lens.

Lastly, suppose (c_x, c_y) and R are the center and radius of circular region in the fisheye image, the transformation of the pixel coordinate (u, v) in the fisheye image is as follows

$$\begin{aligned} u &= c_x + R \times r \times \cos\phi \\ v &= c_y + R \times r \times \sin\phi \end{aligned} \tag{5}$$

C. Image Mosaic

In this paper, the image matching process is based on the Speed Up Robust Features (SURF) [15] and k-Nearest Neighbors (k-NN). SURF detector/descriptor is used to extract regions of interest and compute 64-dimensional descriptors for these regions. The descriptor can be represented as $D = \{q_1, q_2, q_3, q_4, ..., q_{64}\}$, D represents the descriptor vector and q_i is the Harr wavelet response for different directions in different sub-regions. The k-NN algorithm is used to calculate the Euclidean distance between two descriptor vectors of the matching images, and find out the best k nearest points. In order to get stable matches, the matching is performed in two directions. That is to say, for each feature point in one matching image, two closest neighbors should be found in the other image.

RANSAC [16] is used to reject the wrong alignments. Its procedure can be described as following steps: 1) randomly choose four pairs of matched points to calculate the transform matrix *H*; 2) compute the Euclidean distance d_i for the rest of matched points and keep the matched pair as inliers only when d_i is less than a threshold; 3) recalculate the transform matrix *H* using the point set with most inliers and repeat the above steps until it reaches the maximum iteration; 4) compute the transform matrix *H* on the last point set.

Because the original images are produced by three separated cameras, there are always differences of exposure and color, which will result in the splicing traces. Image fusion is needed to eliminate the traces. Similar to [17], multi-resolution image fusion method is adopted. Fig. 3 shows an example of the panoramic image generated from three fisheye images.

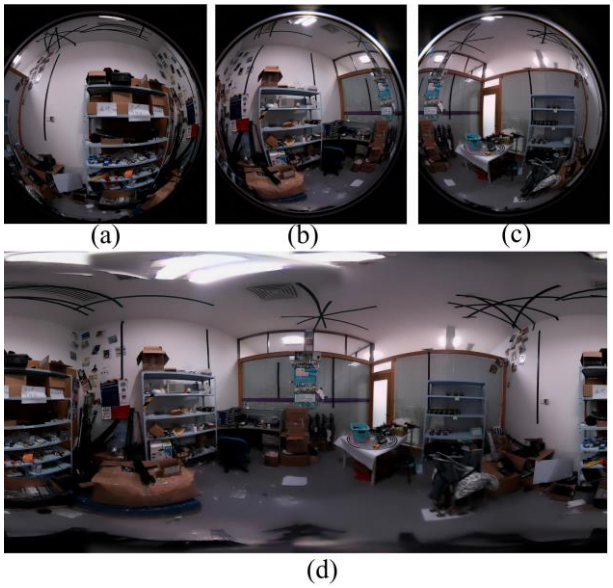


Figure 3. Panoramic image generation, (a), (b) and (c) are the original fisheye images; (d) is the constructed panoramic image.

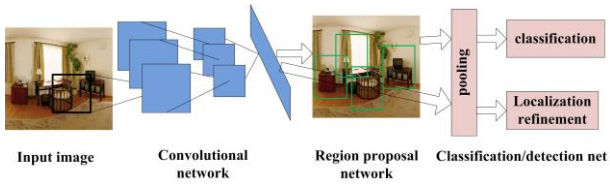


Figure 4. Architecture of object classification and detection network

III. MULTI-CLASS OBJECTES DETECTION

A. *Panoramic Image Dataset*

There is no publicly available panoramic image dataset for object detection to our knowledge. Thus, we have created one by ourselves. At present, we only focus on indoor object detection. 2000 indoor panoramic images (including 700 images from [18]) have been collected from different rooms with varied illumination and view angles. The resolution is 9000×4500.

All the panoramic images have been manually labeled according to the PASCAL VOC format. There are mainly 10 kinds of labeled objects, including bed, sofa, table, monitor, curtain, chair, door, window, picture and mirror. Normally, there are only two classes of objects on average in the PASCAL VOC dataset. However, there are as many as eight classes in each collected panoramic image. This is an obvious advantage for panoramic image.

B. *Object Classification and Detection Network*

In this paper, the object classification and detection on the panoramic image is based on Faster R-CNN [14]. The framework is shown in Fig. 4. The inputs of the network are the image of any size. A CNN network is used to extract feature map. Then the region proposal network (RPN) takes the feature as input and outputs a set of rectangular object proposals. A feature vector will be obtained for each proposal. The normalized feature vector obtained from the spatial pyramid pooling layer is fed into the classification and detection net.

The Faster R-CNN incorporates RPN and Fast R-CNN by sharing a common set of convolutional layers. One of important advantages of this deep network is that it has the capability of detecting objects in a wide range of scales and aspect ratios. In our experiments, we use the Simonyan and Zisserman model (VGG-16) [19] as the shareable convolutional layers. As is shown in Fig. 5, for the CNN network, there are 5 groups of convolutional net consisting of 13 layers and each group is followed by a pooling layer. All the convolutional kernels have the same size of 3×3 and the convolutional stride is 1. For the pooling layer, the kernels have a smaller size of 2×2 with a stride of 2. There are 3 fully connected layers followed by those conv-pooling layers. Both the first and second layers have 4096 dimensions while the third one has 1000 dimensions.

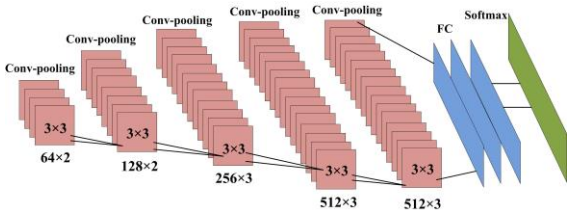


Figure 5. CNN structure (conv-pooling: convolutional layer followed a pooling layer; FC: fully connected layer)

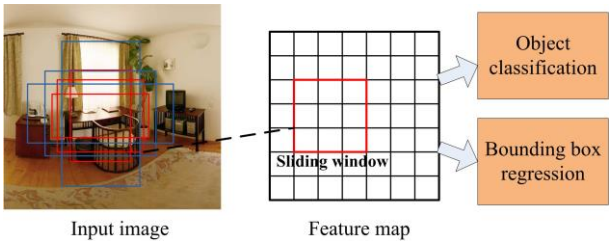


Figure 6. Schematic of region proposal network

TBALE I. REION PROPOSAL SIZE FOR EACH ANCHOR

anchor	64^2 1:1	64^2 1:2	64^2 2:1	64^2 3:2
proposal	64×64	45×90	90×45	86×57
anchor	128^2 1:1	128^2 1:2	128^2 2:1	128^2 3:2
proposal	128×128	90×128	180×90	172×114
anchor	256^2 1:1	256^2 1:2	256^2 2:1	256^2 3:2
proposal	256×256	181×362	362×181	343×229
anchor	512^2 1:1	512^2 1:2	512^2 2:1	512^2 3:2
proposal	512×512	362×724	724×362	457×686

To generate region proposals, a small full convolutional network is connected to the CNN network (Fig. 6). A 3×3 window is sliding over the last convolutional feature map of CNN network. Every window will be mapped into a normalized feature vector which is fed into two separated fully connected layers (one is for object classification and the other for bounding box regression). At each sliding-window location, *n* region proposals (anchors) are simultaneously predicted. In our experiments, 4 scales ({64^2, 128^2, 256^2,

512^2}) and 4 aspect ratios ({1:1, 1:2, 2:1, 3:2}) are used, i.e., the n equals to 16. The average proposal size for each anchor is shown in Table I. These proposals are initial reference boxes for training. After learning, candidate bounding-box can be generated for the object in the image. In order to train the region proposal network, those reference regions should be labeled into positive/negative sample according to the following rules [14]: 1) the region with the highest overlap with true region candidate is labeled positive sample; 2) for the rest regions of step 1), the region with a overlap more than 70% is labeled positive while the region with a overlap less than 30% is labeled negative; 3) the rest and regions crossing the image boundary are not labeled.

The corresponding feature vector should be normalized before it is fed into the subsequent classification/detection net because the size and aspect ratio for the candidate region generated by RPN are very different. In this paper, spatial pyramid pooling [20] is used to normalize feature vector with arbitrary dimension. Three scales ({4×4, 2×2, 1×1}) are adopted to produce pyramids. For very region, max-pooling is used to calculate the feature vector. In the experiments, there are 512 convolutional feature maps generated by the CNN and then the final normalized feature vector with $21\times512=10752$ dimensions is obtained.

The cost function for an image is defined as [14]:

$$L_{\text{total}}=(\sum_i L_{cls}(p_i,p_i'))/N_{cls}+\lambda(\sum_i p_i' L_{reg}(t_i,t_i'))/N_{reg} \quad (6)$$

where the i is the index of sample and p_i is the predicted probability of region proposal i being an object. The ground-truth label p_i' is 1 if the region proposal is positive and is 0 if the region proposal is negative. t_i is a vector representing the position of the predicted bounding-box, and t_i' represents the position of the ground-truth box. N_{cls} and N_{reg} are the size of mini-batch and the number of region proposals. The classification cost L_{cls} is a log cost $L_{cls}(p_i,p_i')=-\log(p_i)$ for true class p_i'. The regression cost L_{reg} is defined as $L_{reg}(t_i,t_i')=R(t_i-t_i')$ where R is a smooth function defined in [13]. λ is a balancing parameter for classification and regression cost.

C. Network Training

Images of a single scale are used to train and test the networks. A statistic analysis has been done on the scales of ~30000 objects in the 2000 panoramic images and it is found the mean width and height for the minimum object is 130 pixels. To reduce the computation, we re-scale the panoramic image into 2500×1250 and ensure the minimum object still has a resolution of about 36×36. An alternating training algorithm similar to [14] is used.

1) The convolutional network is initialized with an ImageNet-pre-trained model. The parameters for the RPN are randomly initialized. The RPN is first trained with a learning rate of 0.001 and a momentum of 0.9. The total number of iteration is 80000.

2) The proposals generated by the step 1) are used to train the detection network by the R-CNN, which is also initialized with ImageNet-pre-trained model. The total number of iteration is 40000.

3) The parameters learned from step 2) are used to initialize the RPN. The shared convolutional network is fixed and only the RPN layers update. It runs for 80000 iterations.

4) The shared convolutional network is still kept fixed and the classification and detection layers are added to form a unified network. 40000 iterations are run to fine-tune the network.

IV. Experimental Results

The network training and testing are implemented by Caffe [21]. The computation uses a CPU (Intel i7) and a GPU (Nvidia Titan X). To expand the image dataset, we put the 2000 panoramic images and 9963 normal images from PASCAL VOC 2007 together. Half of the images are used for training and the rest are for testing. For the detection calculation, only the panoramic images are considered.

A. Effect of Region Proposals

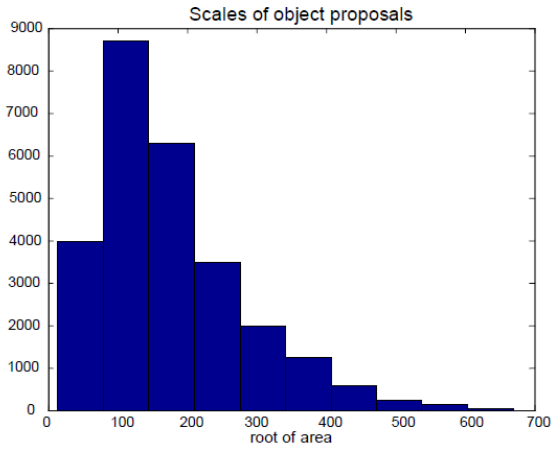


Figure 7. Histograms of the root of area for object

TABLE II. Performance with Different Region Proposals

Scale	Aspect ratio	mAP
{128^2}	{1:1}	63.7%
{256^2}	{1:1}	65.3%
{512^2}	{1:1}	64.2%
{128^2}	{1:1, 1:2, 2:1}	66.7%
{256^2}	{1:1, 1:2, 2:1}	65.9%
{128^2, 256^2, 512^2}	{1:1}	67.3%
{64^2,128^2, 256^2, 512^2}	{1:1, 1:2, 2:1, 3:2}	68.7%

The effect of region proposals on the mean average precision (mAP) is investigated using different groups of scale and aspect ratio for the proposal. As is shown in Table II, it can be roughly concluded that the mAP tends to increase with more scales and more aspect ratios. The

optimal configuration should depend on the objects in the image dataset. For all the objects in the panoramic images, the roots of their areas are calculated and the statistical distribution is shown in histogram (Fig. 7). It is found that most of the object scales fall in the range of 502 to 3002 and only a few are over 5002. Furthermore, the aspect ratios of the objects are also calculated as depicted in Fig. 8. Obviously, most of the objects have an aspect ratio of less than 3. Therefore, four scales ({64^2, 128^2, 256^2, 512^2}) and four aspect ratios ({1:1, 1:2, 2:1, 3:2}) are chosen for the region proposals in our experiments.

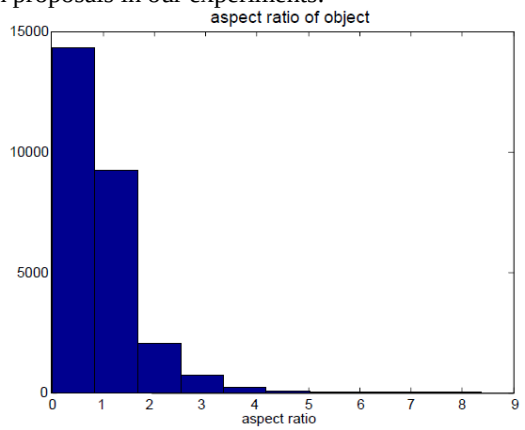
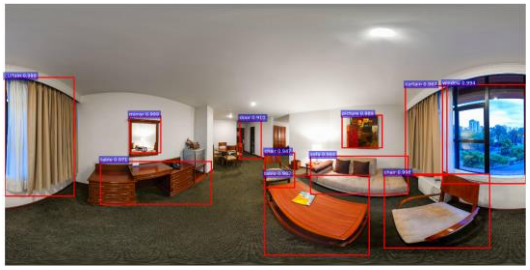
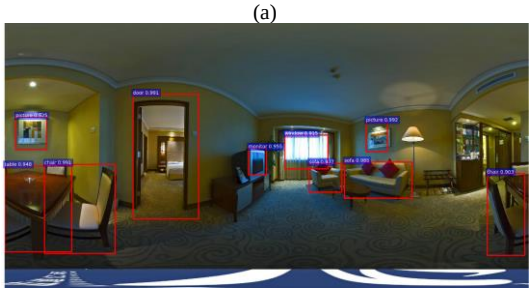


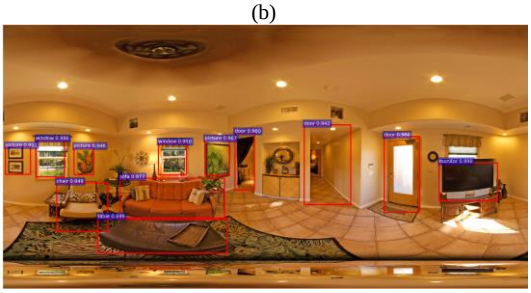
Figure 8. Histograms of aspect ratio



(a)



(b)



(c)

Figure 9. Object detection in panoramic images, (a), (b), (c) are examples for indoor object detection using proposed method.

B. *Test Results*

The Deformable Parts Model (DPM) [22] is another important approach which has also gained a lot of attentions in object classification and detection. This approach uses dimensional-reduced HOG feature and latent SVM for training. In this paper, the DPM is implemented for comparison. The test on 10 categories of objects in the panoramic images is shown in Table III. Some detection examples are demonstrated in Fig. 9. It can be seen from Table III that the detection precision of the DPM is less than 40% for most of the object categories except for picture. Although the DPM-based approach performs very well in detecting normal object without distortion, it does not for the deformed objects in the panoramic images. Because the DPM still depends on the conventional HOG feature which is incapable of fully representing many kinds of distorted objects. The detection of picture is an exception with a much higher precision of 56%. That's because most of pictures appear in the shapes of rectangle and parallelogram in the images. They are not severely distorted due to non-rigid transformation. On the other hand, the pictures contain much additional information such as color and texture. Comparatively, the mirror and window have similar geometric shape with picture but very few salient features. Their detection precisions are only 19.2% and 21.8% respectively and much lower than that of picture.

TABLE III. DETECTION PRECISION

category	**DPM**	**proposed**
bed	35.2%	76.3%
table	21.6%	73.6%
monitor	31%	70%
curtain	29.5%	69.5%
chair	26%	68%
door	31.9%	67.3%
window	21.8%	62.6%
Picture	56%	68%
mirror	19.2%	58.7%
sofa	22.2%	72.5%
mAP	29.4%	68.7%

Contrarily, the R-CNN based approach used in this paper has a much better performance. The mAP for the ten categories is up to 68.7%. Compared to the conventional feature, e.g., HOG, CNN feature belongs to high level features and is very powerful for the representation of deformed or non-deformed object. Furthermore, it is noted that the detection precisions of some categories, such as bed, table and sofa, are higher than 70% while the detection precisions for the window and mirror are about 60%. Part of the reason is that the total number of bed/table/sofa is much higher. It means that detection precision could further increase with increasing samples. This is an important advantage for deep learning. In addition, it takes about 350ms to detect all the objects in each panoramic image using the proposed method while it takes 45s using the DPM in our experiments.

V. CONCLUSIONS

In this paper, it presents an efficient method to create panoramas and build datasets. The proposed method uses a panoramic camera which consists of three fisheye cameras and works in a sharing mode. Furthermore, R-CNN based deep learning is proposed to detect the objects with strong distortions on the panoramic images. For the multi-class objects detection in the panoramas, the region proposal technique is efficiently adaptive to the wide range of scales and aspect ratios given a priori on the objects. The detection precision reaches 68.7% and could be increased with increasing data. It is interesting to build a real-time object detection system based on panoramic camera for the practical applications. For an example, the detection system is capable of virtually showing the specific object in a room which people are concerned about. And that would be our future work.

ACKNOWLEDGMENT

This work is supported by Shenzhen Hi-Technology Funding under Grant No. CXZZ20140419141609644 and GJHZ20150312114457505. This research is also partly supported by NSFC under Grant No.91648102.

REFERENCES

[1] H. Kim, J. Jung and J. Paik, "Fisheye lens camera based surveillance system for wide field of view monitoring," Optik, vol. 127, pp. 5636–5646, 2016.

[2] M. Bertozzi, L. Castangia, S. Cattani, A. Prioletti, P. Versari, "360 ° Detection and tracking algorithm of both pedestrian and vehicle using fisheye images," 2015 IEEE Intelligent Vehicles Symposium , IEEE, June 28-July 1, 2015, pp.132-137.

[3] M. Lin, G. Xu, X. Ren, K. Xu, "Cylindrical panoramic image stitching method based on multi-cameras," The 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE, June 8-12, 2015, pp.1091-1096.

[4] M. Bui, V. Fremont, D. Boukerroui, P. Letort, "Deformable parts model for people detection in heavy machines applications," 2014 13th International Conference on Control, Automation, Robotics & Vision, IEEE, December 10-12, 2014, pp.389-394.

[5] D. Dooley, B. McGinley, C. Hughes, L. Kilmartin, "A Blind-Zone Detection Method Using a Rear-Mounted Fisheye Camera With Combination of Vehicle Detection Methods," IEEE Transactions on Intelligent Transportation Systems, vol.17, no.1, pp:264-278, 2015.

[6] D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Comptuer Vision, vol. 60, no.60, pp. 91-110, 2004.

[7] T. Ojala, M. Pietikainen and T. Maenpaa, "Gray scale and rotation invariant texure classification with local binary patterns", IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 24, no. 7, pp. 971-987, 2002.

[8] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", IEEE Conference on Computer Vision & Pattern Recongnition, IEEE, 2013, pp.886-893.

[9] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast Feature Pyramids for Object Detection," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 36, pp. 1532-1545, 2014.

[10] M. Gabb, O. L ̈ohlein, R. Wagner, A. Westenberger, M. Fritzsche, and K. Dietmayer, "High-performance on-road vehicle detection in monocular images," 2013 16th International IEEE Conference on Intelligent Transportation Systems, IEEE, Oct, 2013, pp. 336–341.

[11] J. Marin, D. Vazquez, A. Lopez, J. Amores, and B. Leibe, "Random forests of local experts for pedestrian detection," IEEE International Conference on Computer Vision, IEEE, Dec, 2013, pp. 2592–2599.

[12] J. Schmidhuber, "Deep learning in neural networks: an overview," Neural Networks, Vol. 61, pp.85-117, 2015.

[13] R. Girshick, "Fast R-CNN", IEEE International Conference on Computer Vision, IEEE, Dec. 11-18, 2015, pp.1440-1448.

[14] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks", IEEE Transactions on Pattern Analysis and Machine Intelligence , vol.PP, no.99, pp.1-1, doi: 10.1109/TPAMI.2016.2577031.

[15] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "Speeded-up robust features", Computer Vision and Image Understanding, vol.110, no.3, pp.346-359, 2008.

[16] M.A. Fischler, R.C. Bolles, "Random sample consensus: a paradiigm for model fitting with applications to image analysis and automated catography", Communication of the ACM, vol. 24, pp.381-395, 1981.

[17] M. Pradeep, "Implementation of image fusion algorithm using MATLAB (LAPLACIAN PYRAMID)," International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing, IEEE, 2013, pp. 165-168.

[18] Y. Zhang, S. Song, P. Tan and J. Xiao, "PanoContext: A whole-room 3D context model for panoramic scene understanding," 13th European Conference on Computer Vision (ECCV), Sep.06-12, 2014, pp. 668-686.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in International Conference on Learing Representations (ICLR), 2015.

[20] K. He, X. Zhang, S. Ren, J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 37(9), pp.1904-16, 2015.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," arXiv:1408.5093, 2014.

[22] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan, "Object detection with discriminatively trained part-based model," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, 2010.