Distance Measurement in Industrial Scenarios

Hao Liu*, Chengzhao Liu¹, Chenzhe Ma¹, Hantao Zheng² and Jianglong Xu²
¹ Power China Zhongnan Engineering Corporation Limited, Changsha, Hunan, 410014, China
² College of Computer Science and Electronic Engineering, HNU, Changsha, Hunan, 410014, China
Email: 841969893@qq.com

Abstract. 3D reconstruction technology is one of the important technologies of computer vision. Compared with 2D data, 3D space contains more abundant information, including location information, local / global features and so on. It is of great significance to solve the problem of target distance measurement in a single image scene. It is difficult for neural network to predict hole size without image scale information. In this paper, we focus on the measurement of objects with little change in height in the image. We use the relationship between camera parameters and regional features and the internal relationship between regional features to solve the problems of three-dimensional parameter reconstruction and monocular image distance measurement and use the standard cross entropy loss to optimize the transformer model. This model has achieved good results on the sampled data set.

1. Introduction

The goal of image-based 3D reconstruction is to infer the 3D geometry of objects and scenes from one or more 2D images. This long-standing ill posed problem is very important for many applications such as robot navigation, target recognition and scene understanding, 3D modelling and animation, industrial control and medical diagnosis. The key point of 3D reconstruction technology is how to obtain the depth information of target scene or object.

Many 3D reconstruction methods are good at using a priori knowledge to describe the 3D reconstruction problem as a recognition problem. This method has high sub dimensions and needs to consume more memory. At the same time, it has some problems, such as high computational complexity and poor real-time performance. Another considerable is the use of deep learning technology and large training data sets. Among them, [1] introduces a new part-based intermediate representation, coupled with a four stage increasingly complex depth neural network, which is the first method to estimate the height only from a single depth image. [2] A CNN based method is proposed to estimate the height of standing children under 5 years old by using the depth images collected by smart phones. For example, the pioneering work “single view measurement” by criisi et al. [3] depends on the size of the reference object in the scene.

In this work, we mainly solve the monocular image obtained under unconstrained conditions to restore the absolute scale of the scene, such as the three-dimensional height of the object, the height of the camera from the ground, the direction of the camera and the field of view parameters. We use a multimodal data processing model. The method mainly includes target detection module and prediction estimation module. Given an input image, our goal is to determine whether there is an object of interest on the barbed wire fence and the size of the object. In fact, we pay more attention to...
the holes on the barbed wire fence, signs and barrels, because these objects are very common on the construction site and the height change is very small. Specifically, we will extract image features and input RPN to generate region proposals. Combined with the camera parameters output by FC, including camera direction (or horizon in the image) and field of view, we will convert any 2D measurement in image space into 3D measurement, as well as the absolute 3D height of the camera from the ground, so as to calculate the size of the object more accurately.

2. Related Work

In recent years, great progress has been made in the research of target detection and image segmentation based on deep learning, and the accuracy and efficiency of target detection and image segmentation have also been significantly improved. The use of deep learning methods to solve the recognition and detection of 3D target table, object attitude estimation and 3D modeling have also attracted more and more researchers' attention. In order to solve the unavoidable problems in traditional 3D modeling, such as multiple identification, reconstruction, capture, iteration and recapture of real areas. In the research process, Izadinia et al. [4] detects the key areas, then looks for similar shapes from the database, speculates the scene structure, and adjusts the pose and size of the three-dimensional object to better match the synthetic rendering structure with the actual scene. Tulsiani et al. [5] uses synthetic rendered images and their related 3D scenes for training. In this method, the encoder and decoder are used to estimate the scene layout, and the low-resolution imaging of the scene is mapped to a hidden space. Finally, the is pooled to obtain the pooled features. Finally, the decoder is used to generate the voxel grid of the object, and the three-dimensional scene is analyzed in the form of position and proportion. Tianwei Zhang et al. [6] proposed using the dynamic segmentation of optical flow residuals and dense fusion RGB-D SLAM method for scene reconstruction. In this method, by improving the influence of dynamic factors, the corresponding dynamic image is extracted from the current effective frame, and the real environment is reconstructed by analyzing the image.

With the in-depth research on visual odometry, in order to pursue higher efficiency and accuracy, more and more people combine research methods with deep learning [7-8] to obtain further research results. In previous work, CNN has to be used to fit pose and keep contact in time. Some methods use parallax between frames to keep contact, in addition, some researchers use optical flow to carry out research work. Among them, Huang Zhan et al. [9] first explored the method of deep learning and the combination of epipolar geometry and perspective-n-point, and proposed a frame by frame visual odometry algorithm, so as to avoid the problem of scale drift assisted by single view depth CNN with consistent scale, and greatly improve the robustness of the algorithm. You-Yi et al. [10] in order to solve the shortcomings of traditional VO algorithm in robustness, accuracy and scalability, so as to directly optimize the geometric attitude of the target. Thus, an end-to-end trainable framework including feature extraction, matching and interference point deletion is designed.

For a long time, non-contact object size measurement has been the focus and difficulty of researchers. Fukun Yin et al. [1] and others proposed a method to measure human body data using a single image. In this method, they only use the depth data of the image and use FCN [11] to segment the human head, upper body, thigh and lower leg. Then the height is predicted by using the relationship between them. Bovyrin et al. [12] proposed a method to predict the object size by using the transformation between the camera and the 3D model to obtain the parameters to estimate the actual size of the object. Lee et al. [13] project the two-dimensional features of the detected object into the three-dimensional space, set the coordinate system of the object to be detected, and then use the relevant feature points of the object to be detected to constrain and modify the measurement process, so as to realize the object size measurement of a single image.
3. Method
This paper studies the object distance measurement in single image scenario. In this section, we will discuss the output of the model and how to reconstruct 3D parameters and estimate the length of the object to be measured from the model output.

Specifically, what we need to do is to identify whether there are holes in the chain link fence, and the size of the holes. Under the condition of no image scale information, it is difficult for neural network to estimate the size of the hole information. Therefore, in addition to the holes in the chain link fence, we also set the signs and buckets as our object of interest which are very common on the construction site and have little variation in height, by detecting the height of common objects and comparing their depth in the image, we can measure the size of holes.

We will discuss in detail the mathematical principles and derivation process involved in the above process, as well as the optimization strategy of the model.

3.1. Problem Formulation
This paper deals with the ranging problem of monocular images, that is, input an image and measure the width or height of the object of interest in the image.

To simplify things, let's assume that the camera only has a vertical tilt angle (VTA) $\theta_{vta}$ to the ground when taking a picture, and zero roll (the horizon line in picture is horizontal and we call the projection of the vanishing line of the ground into the image the horizon). We define the center position of the input image as $(x_c, y_c)$, the pixel size of image is $H \times W$. Given an input image, the model will output camera parameters: vertical field of view (VFOV) $\theta_{vfov}$, VTA, camera height $h_{cam}$, and objects information: $(h_i, c_i, d_i, x_i, y_i, w_i, h_i)$ where $i$ means the $i$-th object, $h$ means height, $c$ means class, $d$ means depth, $x$ and $y$ means the center position of the object's boundary box in image and $w$ and $h$ are the shift of boundary from center. From the $\theta_{vfov}$, we can calculate the focal length $f$ of the camera as:

$$f = \frac{1}{\tan\left(\frac{1}{2} \theta_{vfov}\right)} \tag{1}$$

and get the horizontal field of view (HFOV) $\theta_{hfov}$ as:

$$\theta_{hfov} = 2 \arctan\left(\frac{W}{f}\right) \tag{2}$$

3.2. Model Architecture
The architecture of our multi-modal model is illustrated in figure 1. In general, it includes object detection module, camera parameters estimation module and a RoI depth predicting module. We use Faster R-CNN [14] to detect objects of interest.

3.2.1. Feature Extractor. For an input image $I$, it will be fed into the feature extractor, we define it as:

$$x = \phi(I) \tag{3}$$

where $\phi$ is the feature extractor and $x$ is the feature map of $I$. In this work, we use ResNet-50 [15] as the feature extractor. For an image whose input size is $W \times H$, we get its corresponding feature map $x \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 1024}$.

3.2.2. Region Proposal Network. Then, taking $x$ as the input of RPN. The RPN generates a set of bounding box predictions $\{box_1, box_2, \ldots, box_n\}$. where $box_i = \{x, y, w, h\}$ indicating the position and size of box, specifically, $x, y, w,$ and $h$ denote the two coordinates of the box center, width, and height. However, RPN only generates regional proposals, and these regions are not precise. Therefore, we use pooling layer to get feature maps of these regions with the same size and input these feature
maps into subsequent networks to get the final detection results. In this work, we add a height prediction head to regress the height of the detected objects.

3.2.3. Camera Parameters Recovery Module. Feature images contain deep information of the original image, which is sensed by the fully connected layers and predicted, and three linear regression heads are used to predict camera height, VFOV and VTA.

**Figure 1.** Model overview. Left: We used Faster R-CNN to detect regions of interest (RoI) in the image. It should be noted that we added a full connected layer to predict the real-world height of the region. Dashed lines indicate that the gradient will not propagate during training. Right: We design a transformer-based network to predict the depth of detected objects, \( R_i \) comes from the output of ROI Pooling.

3.2.4. Depth Prediction Network. Recent work [16] has used LSTM RNN [17] to predict the depth of a moving target. Activated by this and considering the efficiency of the transformer module [18], we design a transformer-based model (figure 1 Right) that leverages the relationship between camera parameters and region features and the intra-relationship between region features to predict the depth of detected objects. In detail, all the inputs will be embedded to the same dimension in the embedding block, for the feature map \( x \in \mathbb{R}^{h \times w \times d_f} \), we define the embedding \( x_e \in \mathbb{R}^{d_e} \) as:

\[
x_e = \sigma \left( \text{proj}(\text{flat}(x)) \right)
\]

where \( d_e \) is the uniform embedding dimension, \( \text{flat} \) is the reshaping process, \( \text{proj} \) is a linear layer and \( \sigma \) is the activation function. We then treat regional image features in the same way. As for camera height, VFOV and VTA, we use a 1D convolution layer and a pooling layer to embed these. These embeddings will be fed into the Transformer Block which outputs \( V \in \mathbb{R}^{n \times d} \), where \( n \) is input length and \( d \) is the output dimension. Then we use FC layers and three independent FC heads to predict the depth. However, because input to the location of the additional parameters like VFOV also results in depth predictions, we must discard them, which is what the Result Filtering component does. Finally, we get the depth prediction of each detected objects \( D = \{ d_1, d_2, \ldots, d_n \} \in \mathbb{R}^{n \times 1} \), where \( n \) is the number of detected objects.

3.3. Optimization Strategy

The overall objective is to minimize the following loss \( L \):

\[
L = L_{\text{cls}} + L_{\text{obj}} + L_{h_{\text{obj}}} + L_{d_{\text{obj}}} + L_{\text{Cam}}
\]

where \( L_{\text{obj}} \) is the detection loss, \( L_{\text{cls}} \) is the classification loss, \( L_{h_{\text{obj}}} \) is the height prediction loss, \( L_{\text{Cam}} \) is the camera parameters recovery loss, \( L_{d_{\text{obj}}} \) is the depth prediction loss.
3.3.1. Detection Loss. We use GIoU Loss [19] as $L_{obj}$ which defined as:

$$L_{obj} = IoU - \frac{\left| C \right|}{\left| A \cup B \right|} \tag{6}$$

$$IoU = \frac{\left| A \cap B \right|}{\left| A \cup B \right|} \tag{7}$$

where $A$ is the prediction boundary box, $B$ is the ground truth, $C$ is the smallest box enclosing $A$ and $B$.

3.3.2. Classification Loss. We use a standard cross entropy loss to optimize the model to ensure that the model can accurately classify the objects identified.

3.3.3. Camera Parameters Recovery Loss. We define the camera parameters $p = \{\text{height, vfov, vta}\}$ and $\hat{p}$ is the model prediction, then we define $L_{\text{Cam}}$ as:

$$L_{\text{Cam}} = ||\hat{p} - p|| \tag{8}$$

3.3.4. Height Prediction Loss. Similar to [20], we set gaussian prior loss for common objects, then the $L_{h_{obj}}$ is defined as:

$$L_{h_{obj}} = -\left( \sum_{i=1}^{N} \frac{1}{\mathbb{I}(y_i = 1)} \sum_{i=1}^{N} \mathcal{P}(\hat{h}_i; \mu_1, \sigma_1) \mathbb{I}(y_i = 1) + \sum_{i=1}^{N} \frac{1}{\mathbb{I}(y_i = 2)} \sum_{i=1}^{N} \mathcal{P}(\hat{h}_i; \mu_2, \sigma_2) \mathbb{I}(y_i = 2) \right) \tag{9}$$

where $y_i$ is the object class, in this work, we set tow common classes and corresponding labels are 1 and 2. $N$ is the number of detected objects.

3.3.5. Depth Prediction Loss. The $L_{d_{obj}}$ can be defined as:

$$L_{d_{obj}} = \frac{1}{N} \sum_{i=1}^{N} (\hat{d}_i - d_i) \tag{10}$$

Where $\hat{d}_i$ is the predicted depth of $i$-th object and $d_i$ is the ground truth.

3.4. Size Measurement of Target Objects

In this work, our goal is to detect and measure the size of holes in the chain link fence. The trained model does a good job of detecting holes and common objects and predicting their depth and the height of common objects. With above information, we can measure the size of holes. Using only information from a common object, we can calculate the height $h$ of the hole as:

$$l_1 = \frac{d}{\sin \left( \frac{B_{\text{vfov}} |x_i - x_c|}{4H} \right)} \tag{11}$$

$$l_2 = \frac{d}{\sin \left( \frac{B_{\text{vfov}} |y_i - y_c|}{4H} \right)} \tag{12}$$

$$l = \left( l_1^2 + l_2^2 \right)^{1/2} \tag{13}$$
we define the \( I \) with data of common object as \( I^{\text{common}} \), and \( I^{\text{target}} \) for target object data. Then we get \( h^{\text{target}} \):

\[
h^{\text{target}} = \frac{h^{\text{common}}}{l^{\text{common}}l^{\text{target}}} \tag{14}
\]

where \( h^{\text{common}} \) is the height prediction of common object from model. And We can compare the target object to each common object, calculating the corresponding \( h \), finally, the target object height is obtained by averaging. In the same way we can get the width of target object.

4. Experiment

Due to the lack of datasets of building site and in order to comprehensively evaluate the effect of our model, we move the problem formulation to a general scenario, that is to say we trained our model on several commonly used datasets and compared the effect with other baseline model. We treat the people's height as the interest object since people's height are relatively uniform, then we take the height predicted by the model for other objects as the standard to evaluate the accuracy of the model.

To check the effect of the model in our problem formulation, we also use a private building site dataset to check the performance of our model and baselines.

4.1. Datasets

4.1.1. KITTI. KITTI [21] is the largest computer vision algorithm evaluation data set in the automatic driving scene in the world, it provides comprehensive camera parameters and target position estimation. To meet our problem formulation, we pick up samples with pedestrian from the dataset. Specifically, we use Mask R-CNN to filter out samples that do not meet the requirements of the problem formulation, keep only those samples that contain the whole person. By this way, we yield a dataset which contains 10,000 images, includes 2500 images for validation and 500 images for testing.

4.1.2. SUN360. In order to train the camera parameter estimation ability of the model, we follow the method of [22] and pick up the panorama with camera parameters from the SUN360 database [23]. We divide the screening results into two parts: 30000 images for training and 3000 images for verification.

4.1.3. UBS. Unsafe Construction Site is the private dataset obtained by taking photos from the real construction site. We label samples manually since there is no ready-made annotation tool. We labeled 161 photos, of which 131 for training and 30 for validation.

4.2. Baseline

We take [24] as the baseline method, to enhance this method, we employ top predictions from Mask R-CNN as input of this method. Based on this method, we set two baseline models: (1) PGM: A Probabilistic Graphical Model which accepts object recommendations and surface geometry and predicts camera height and horizon. (2) PGM-fixedH: Same as PGM, but all objects are assumed to have normal height, here we set 1.7m as the people's normal height.

4.3. Training

Since the parameters of Camera Parameters Recovery Module are difficult to converge, we take a two-stage training method on our model. In first stage, we trained Camera Parameters Recovery Module solely to make model converge faster in the second stage. We use the SUN360 dataset training model in first since it provides ground true camera parameters.

In the second stage, we train the model on KITTI dataset and UBS dataset, since both of them provide object parameters and camera parameters, we can train the Camera Parameters Recovery Module and target detection module.
4.4. Result
We directly evaluate the errors in camera height estimation \( \epsilon_{h_{\text{Cam}}} \) and object height estimation \( \epsilon_{h_{\text{obj}}} \), the detailed definitions of symbols are as follows:

\[
\epsilon_{h_{\text{Cam}}} = \left\| h_{\text{Cam}} - h_{\text{Cam}_{\text{gt}}} \right\| \tag{15}
\]

\[
\epsilon_{h_{\text{obj}}} = \frac{1}{N} \sum_{i=1}^{N} \left\| h_{\text{obj}} - h_{\text{obj}_{\text{gt}}} \right\| \tag{16}
\]

where \( h_{\text{Cam}} \) and \( h_{\text{obj}} \) are the final estimated camera height and object height, and \( h_{\text{Cam}_{\text{gt}}} \) and \( h_{\text{obj}_{\text{gt}}} \) their ground truth values respectively.

Our method shows better performance than previous work, the structure of our framework makes our model have better effect, table 1 shows the result on the dataset KITTI.

| Table 1. Evaluation errors on KITTI dataset. Best results in bold (lower is better). |
|--------------------------------------------------|-----------------|-----------------|-----------------|
| Error    | Index | PGM-fixed | PGM             | Ours            |
| \( \epsilon_{h_{\text{Cam}}} \)    | mean   | 0.8619    | 0.1402          | **0.0803**      |
|        | std.   | 0.0565    | 0.1477          | 0.0623          |
|        | med    | 0.0790    | 0.0934          | **0.0715**      |
| \( \epsilon_{h_{\text{obj}}} \)    | mean   | 0.1458    | 0.1458          | **0.1266**      |
|        | std.   | 0.1583    | 0.1583          | 0.1154          |
|        | med    | 0.1124    | 0.1124          | **0.0898**      |

We also evaluate our model on the UBS, we chose the same evaluation criteria as what we implemented on KITTI, the result is show in table 2.

| Table 2. Evaluation errors on UBS dataset. Best results in bold (lower is better). |
|--------------------------------------------------|-----------------|-----------------|-----------------|
| Error    | index | PGM-fixed | PGM             | Ours            |
| \( \epsilon_{h_{\text{Cam}}} \)    | mean   | 0.9724    | 0.1825          | **0.1128**      |
|        | std.   | 0.0759    | 0.1733          | 0.0871          |
|        | med    | 0.0883    | 0.1045          | **0.0784**      |
| \( \epsilon_{h_{\text{obj}}} \)    | mean   | 0.1786    | 0.1796          | **0.1536**      |
|        | std.   | 0.1852    | 0.1775          | 0.1358          |
|        | med    | 0.1366    | 0.1378          | **0.1194**      |

5. Conclusion
In this paper, we discuss the output of the target distance measurement model in a single image scene in detail, and study how to reconstruct the three-dimensional model parameters from the model output and estimate the length of the object to be measured. Facing the problem of a single image, we naturally cannot directly obtain the data such as the camera height and field of view parameters of the image, which greatly increases the difficulty of target size measurement. In order to solve this problem, we not only pay attention to the object to be detected, but also detect the height of common objects and compare their depth in the image, the common objects in the corresponding scene are also used as the objects we are interested in for auxiliary detection. In addition, we design a transformer-based model, which uses the relationship between camera parameters and regional features and the internal relationship between regional features to predict the depth of the detected object. In the experiment, we take the protective net, fence, mark and hole on the barrel as the measurement object. The experiment shows that our method has good measurement effect in the corresponding scene in the construction site.
References

[1] Yin F & Zhou S 2020 Accurate estimation of body height from a single depth image via a four-stage developing network Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 8267-8276.

[2] Trivedi A, Jain M, Gupta N K, Hinsche M, Singh P, Matiaschek M, et al. 2021 Height estimation of children under five years using depth images arXiv: 2105.01688.

[3] Criminisi A, Reid I & Zisserman A 2000 Single view metrology International Journal of Computer Vision 40(2): 123-148.

[4] Izadinia H, Shan Q, & Seitz S M 2017 Im2cad Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 5134-5143.

[5] Tulsiani S, Gupta S, Fouhey D F, Efros A A, & Malik J 2018 Factoring shape, pose, and layout from the 2d image of a 3d scene Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 302-310.

[6] Zhang T, Zhang H, Li Y, Nakamura Y, & Zhang L 2020 May Flowfusion: Dynamic dense rgb-d slam based on optical flow 2020 IEEE International Conference on Robotics and Automation (ICRA): 7322-7328.

[7] Wang S, Clark R, Wen H, & Trigoni N 2017 May Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks 2017 IEEE International Conference on Robotics and Automation (ICRA): 2043-2050.

[8] Long J, Shelhamer E, & Darrell T 1997 Long short-term memory Neural Computation 9(8): 1735-1780.

[10] Hochreiter S, & Schmidhuber J 1997 Long short-term memory Neural Computation 9(8): 1735-1780.

[11] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez N, ... & Polosukhin I 2017 Attention is all you need Advances in neural information processing systems: 5998-6008.

[12] Rezatofghi H, Tsoi N, Gwak J, Sadeghian A, Reid I, & Savarese S 2019 Generalized intersection over union: A metric and a loss for bounding box regression Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 658-666.
[20] Zhu R, Yang X, Hold-Geoffroy Y, Perazzi F, Eisenmann J, Sunkavalli K, & Chandraker M 2020 Single view metrology in the wild Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020: Part XI 16, 316-333.

[21] Geiger A, Lenz P, Stiller C, & Urtasun R 2013 Vision meets robotics: The kitti dataset The International Journal of Robotics Research 32(11): 1231-1237.

[22] Hold-Geoffroy Y, Sunkavalli K, Eisenmann J, Fisher M, Gambaretto E, Hadap S, & Lalonde J F 2018 A perceptual measure for deep single image camera calibration Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 2354-2363.

[23] Xiao J, Ehinger K A, Oliva A, & Torralba A 2012 June Recognizing scene viewpoint using panoramic place representation 2012 IEEE Conference on Computer Vision and Pattern Recognition: 2695-2702.

[24] Hoiem D, Efros A A, & Hebert M 2008 Putting objects in perspective International Journal of Computer Vision 80(1): 3-15.