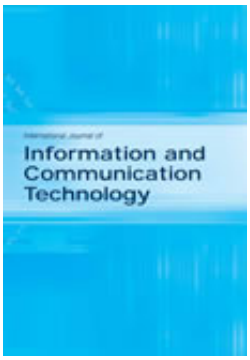
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

SynthBendText3D: a framework for generating scene text data in arbitrary orientations using a 3D graphics engine

Zhao Guan, Weilong Zhang

Article History:

Received:	08 October 2024
Last revised:	24 October 2024
Accepted:	24 October 2024
Published online:	20 January 2025

SynthBendText3D: a framework for generating scene text data in arbitrary orientations using a 3D graphics engine

Zhao Guan* and Weilong Zhang

School of Cyberspace Security and Computer,
Hebei University,
Baoding – 071000, China
Email: guanzhao@stumail.hbu.edu.cn
Email: zhangweilong@stumail.hbu.edu.cn
*Corresponding author

Abstract: To address the domain distribution mismatch between synthetic scene text data and real-world scene text data in arbitrary orientations, we introduce SynthBendText3D – a framework based on a 3D graphics engine that synthesises scene text data in various orientations. The framework generates a large number of text instances in arbitrary directions and constructs a 3D scene to position these instances. By leveraging domain randomisation techniques, it randomises scene parameters such as object arrangement, materials, lighting, and camera angles, ensuring a high degree of diversity in the synthesised data. Moreover, the framework incorporates a polygon reconstruction algorithm to annotate each synthesised text instance with polygonal bounding boxes. Experimental results demonstrate the effectiveness of the data generated by our framework.

Keywords: scene text detection; synthetic data; domain randomisation; domain adaption.

Reference to this paper should be made as follows: Guan, Z. and Zhang, W. (2025) 'SynthBendText3D: a framework for generating scene text data in arbitrary orientations using a 3D graphics engine', *Int. J. Information and Communication Technology*, Vol. 26, No. 1, pp.38–54.

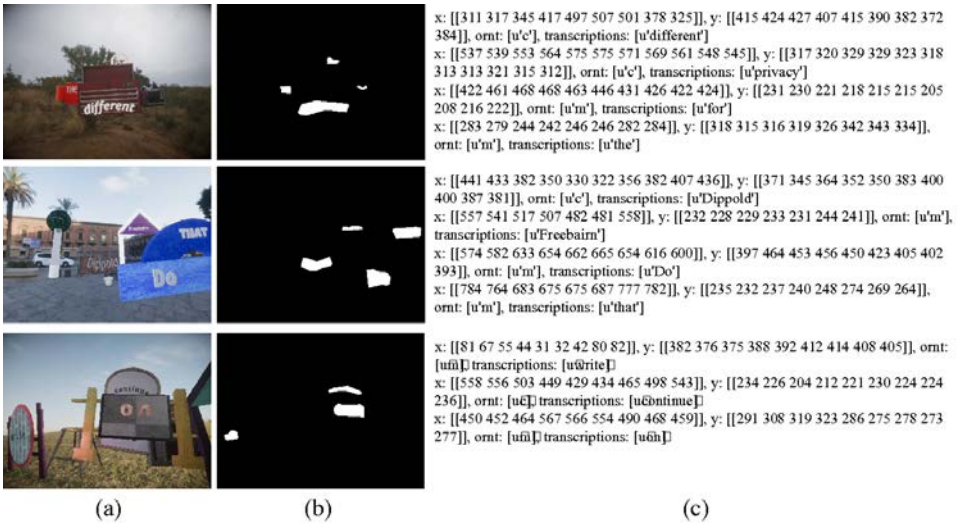
Biographical notes: Zhao Guan received his BS in Digital Media Technology from the Fujian Normal University, Hebei, in 2022. He is currently pursuing his Master's degree with Hebei University. His research interests include computer vision and scene text detection.

Weilong Zhang received his BS in Network Engineering from the Jimei University, Xiamen, in 2022. He is currently pursuing his Master's degree with Hebei University. His research interests is pattern recognition.

1 Introduction

Scene text detection has broad applications in fields such as real-time translation (Lu et al., 2011), autonomous driving (Janai et al., 2020), and robot navigation (Mavrogiannis et al., 2023). However, due to the complexity and diversity of both text styles and natural scene environments, scene text detection remains a challenging problem in computer vision. Deep learning has become the dominant approach in scene text detection, and training such detectors typically requires a large amount of data. Acquiring a significant volume of manually labelled real-world data not only incurs high labor costs but is also prone to annotation errors.

Figure 1 An example of data synthesised by SynthBendText3D, (a) data image (b) binary mask map of the text area (c) data annotations, including polygon coordinate point text boxes, text direction attributes, and text content (see online version for colours)



Synthetic data offers a convenient way to provide large volumes of precisely annotated information. Introducing synthetic data as a supplement to real-world data is a promising solution to address the issue of insufficient data (Mumuni et al., 2024). Most scene text detectors currently rely on synthetic scene text data for pre-training to improve their performance. There is experimental evidence showing that synthetic scene text data is effective for training scene text detectors. Typically, the synthetic data used for training models is considered the source domain, while the real data used for evaluation is considered the target domain, and discrepancies between these domains are inevitable. Existing methods [such as SynthText (Gupta et al., 2016), VISD (Zhan et al., 2018), and SynthText3D (Liao et al., 2020a)] for synthesising scene text data are limited to generating text instances in horizontal or slanted orientations and can only provide quadrilateral bounding box annotations for text instances. However, as the complexity of real-world applications increases, detecting horizontally or slanted scene text is no longer sufficient. In real-world datasets such as Total-Text (Ch'ng and Chan, 2017) and SCUT-CTW1500 (Yuliang et al., 2017), the text appears in arbitrary orientations, and text instance annotations are provided in the form of polygonal bounding boxes. This

indicates a domain gap between the synthetic scene text data and real-world datasets containing text in arbitrary orientations.

A key focus in the current methods of generating synthetic scene text data is determining appropriate regions to place text instances based on scene information to simulate real-world scene text images. Research has shown that compared to placing text instances on 2D static images (Gupta et al., 2016; Zhan et al., 2018), methods based on 3D graphics engines, which construct 3D scenes and accurately simulate real-world parameters, provide more precise scene information, allowing text instances to be placed in semantically coherent locations (Liao et al., 2020a). The use of 3D graphics engines for synthesising data has become quite popular. Studies in synthesising data for other tasks have also demonstrated that using 3D graphics engines in the synthetic data domain is feasible and effective (Lee et al., 2023; Zhang et al., 2022; Abou Akar et al., 2024; Martinez-Gonzalez et al., 2021; Mousavi et al., 2020).

In summary, we propose SynthBendText3D, a framework for generating scene text data in arbitrary orientations using a 3D graphics engine, filling the gap left by previous work that did not support the synthesis of scene text data in arbitrary orientations. Specifically, the framework first synthesises a large number of arbitrarily oriented text models based on Blender, by using modifiers controlled with random parameters. Then, the framework constructs a 3D scene in Unity, importing the synthesised text models and other assets, assigning appropriate regions in the scene, and placing the text models according to specific rules. Using domain randomisation techniques provided by the Unity Perception (Borkman et al., 2021) plugin, parameters such as lighting, material properties, and transformations are randomised to ensure sufficient diversity in the synthesised images. Each text instance in the scene is accompanied by complete segmentation mask annotations. Finally, the framework integrates a polygon reconstruction algorithm to convert the segmentation mask annotations into polygonal bounding boxes for each arbitrarily oriented text instance, ensuring that the annotations are consistent with the format used in the Total-Text dataset (Ch'ng and Chan, 2017). The final dataset generated by SynthBendText3D includes images, binary mask images of text regions, polygonal bounding box annotations, text orientation attributes, and text content labels. Several examples of the synthesised data are shown in Figure 1.

2 Related work

2.1 Scene text detection

The task of scene text detection involves detecting textual information from natural scene images. Early scene text detectors could only detect text instances in horizontal or slanted orientations. Liao et al. (2017) proposed TextBoxes, which uses a single deep neural network to output text regions, including bounding boxes and confidence scores. This method performs well in most cases but struggles with challenging situations, such as detecting overexposed images or those with large character spacing. Zhou et al. (2017) proposed EAST, one of the most classic scene text detectors. EAST utilises a fully convolutional neural network to directly predict text bounding boxes and character-level confidence scores, avoiding some of the complex steps found in traditional scene text detection methods, thereby improving both the efficiency and accuracy of detection.

With the emergence of more complex datasets, such as Total-Text by Ch'ng and Chan (2017) and SCUT-CTW1500 by Yuliang et al. (2017) which contain text instances in arbitrary orientations, represented by polygonal coordinates, the focus of research in the scene text detection field has shifted towards detecting such complex text data. Current scene text detectors can be broadly categorised into regression-based methods and segmentation-based methods. Zhu and Du (2018) proposed SLPR, which enhances the faster R-CNN/R-FCN framework by adding regression of the vertical/horizontal coordinates of polygon intersections along the x/y axis, resulting in polygonal text regions described by 14 points. Zhu et al. (2021) later introduced FCENet, which applies Fourier contour embedding to represent curved text instances. Liao et al. (2020b) proposed DBNet, which uses a differentiable binarisation technique for image segmentation to generate corresponding polygonal text regions. Due to its fast detection speed and outstanding performance, DBNet has been widely applied in real-world scenarios.

As with most deep learning tasks, the lack of sufficient data is a significant bottleneck in improving scene text detection performance. Thus, many scene text detectors incorporate synthetic data for pre-training to ensure their performance.

2.2 Synthetic data based on 3D graphics engines

With the advancement of computer graphics, 3D graphics engines have become capable of rendering images that are closer to reality. Constructing 3D scenes and adding randomised objects, lighting, materials, and other properties to synthesise large amounts of data for training neural networks is a popular method in the synthetic data domain (Nikolenko, 2021). This approach has been proposed and validated for its effectiveness in various computer vision tasks. The Synthehicle dataset, synthesised by Herzog et al. (2023), is used in the field of autonomous driving. Toro et al. (2024) synthesised a dataset using a large number of CAD models based on Blender, applicable in the field of 3D reconstruction. Abou Akar et al. (2024) proposed SORDI.ai, a large synthetic industrial image dataset based on NVIDIA Omniverse.

To make it easier for users to synthesise data using 3D graphics engines, more specialised solutions have begun to emerge, such as BlenderProc (Denninger et al., 2019), BlendTorch (Heindl et al., 2021), Unity Perception (Borkman et al., 2021), and UnrealCV (Qiu et al., 2017). These tools provide programmable and flexible platforms, enabling users to synthesise realistic and diverse virtual data more conveniently. The SAVED dataset proposed by Kim et al. based on UnrealCV (Kim et al., 2022) contains synthetic vehicle images with movable parts, suitable for vehicle-related recognition tasks. Unity Technologies used Unity Perception to create the PeopleSansPeople dataset, designed for human-centric computer vision tasks (Ebadi et al., 2021). Herzog et al. (2023) introduced Synthehicle, a dataset used for vehicle tracking tasks, where scenes were created using CARLA in eight pre-designed town maps, and traffic scenes were recorded using RGB, depth, and semantic LIDAR sensors.

2.3 Synthetic scene text data

Several methods for generating synthetic scene text data have been proven effective in the field of scene text detection. Gupta et al. (2016) introduced SynthText, which blends

text into specific regions of real background images by estimating the depth of the background. Zhan et al. (2018) proposed VISD, which emphasises semantic coherence and visual saliency, using semantic segmentation and saliency maps to determine the placement of embedded text. The model adaptively learns the characteristics of real-world scene text images to determine the colour and brightness of the embedded text. However, the method of embedding text directly into 2D images used by SynthText and VISD can lead to inaccurate perspective relations of the text within the scene. In response to this, Liao et al. (2020a) introduced SynthText3D, which, based on the Unreal Engine, calculates suitable areas for placing text instances in 3D scenes according to surface normals. This was the first study using 3D graphics engines to synthesise data for scene text detection, demonstrating the feasibility of using such synthetic data to enhance the performance of scene text detectors.

However, the aforementioned methods only support generating horizontally or slanted scene text data, which overlooks the diverse orientations of text instances in real-world scenes. Unlike them, our proposed method supports generating scene text data in arbitrary orientations.

3 Structure

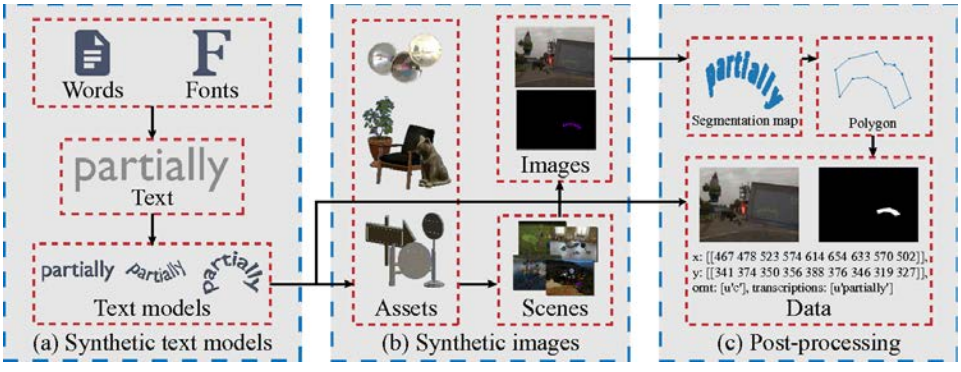
We propose a synthetic scene text data framework named SynthBendText3D, which is designed based on Blender, Unity, and Unity Perception (Borkman et al., 2021). The framework has the following characteristics:

- *Modularity:* The framework is divided into independent modules, each of which can be tested and maintained separately.
- *Flexibility:* Users can adjust the selected assets or related parameters according to the problem requirements, allowing control over the appearance and distribution of the synthesised scene text data (such as the scene of the image, the proportions of text forms, etc.).
- *Realism:* By using a 3D graphics engine for realistic rendering, the synthesised images closely simulate real-life conditions such as uneven lighting and complex environments in scene text scenarios.
- *Diversity:* The synthesised data exhibits sufficient variation, with text instances appearing in arbitrary orientations. For neural networks, this kind of synthetic data reduces the domain gap between synthetic and real data.

The pipeline of SynthBendText3D is illustrated in Figure 2. Specifically, the framework is divided into three modules: synthetic text models, synthetic images, and post-processing. In the synthetic text models module, a large number of text models are generated using Blender, with specific attributes assigned to the text models according to a predefined ratio, forming text models in arbitrary orientations. The synthetic images module utilises Unity's high definition render pipeline (HDRP) to render images by placing the text models in specific locations within a 3D scene. With the assistance of Unity Perception, the generated images are annotated with corresponding segmentation maps of the text instances. In the Post-processing module, based on the segmentation maps generated in the previous module, a polygon reconstruction algorithm is designed

to convert the segmentation maps into polygonal bounding box annotations, resulting in the final dataset.

Figure 2 The pipeline of SynthBendText3D (see online version for colours)



Notes: It consists of three modules: synthetic text models, synthetic images, and post-processing.

3.1 Synthetic text models

This module synthesises a large number of text models in arbitrary orientations for use in subsequent modules.

In this module, text objects are created in bulk using Blender, with random adjustments made to their 'extrude' and 'bevel_depth' properties to give these text objects a small amount of thickness and beveling. Modifiers such as 'twist', 'bend', 'taper' and 'stretch' are added. Based on the proportion of curve-oriented text in the Total-Text dataset (Ch'ng and Chan, 2017), some text models are designed to have a certain curvature effect. The angles of the modifiers follow a truncated normal distribution (Burkardt, 2014). The synthesised models are annotated with information such as text content and text orientation, as shown in several examples of the synthesised text models in Figure 3. The synthesised text models exhibit more diverse orientations compared to the text instances in other works.

Figure 3 Examples of text models

The text content for the created text models is randomly selected from words in the Newsgroup20 dataset, and the fonts used are randomly selected from Google Fonts.

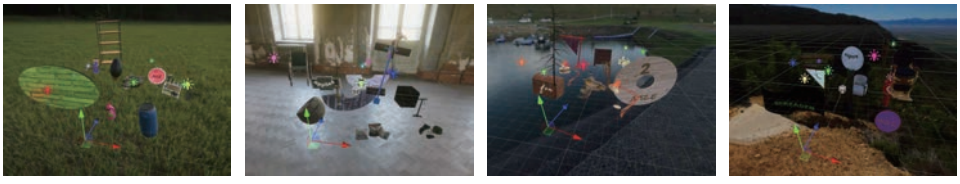
3.2 Synthetic images

This module constructs 3D scenes in Unity, placing the text models at specific locations within the 3D scene to render images of the synthesised data. Image realism and domain

randomisation are considered effective and general methods in the field of synthetic data (Abou Akar et al., 2024), and this module is designed based on these principles.

To ensure the realism of the synthesised data, a large number of high-quality textured 3D models and HDRI maps are collected from the Poly Haven public 3D resource library, supplemented with a certain number of modelled 3D objects imported into Unity, ensuring that these resources do not contain text instances. The module uses Unity's HDRP to synthesise images, allowing the generated images to accurately simulate material textures found in real environments. Box colliders are used to mark several areas on the surfaces of certain models, where text objects are placed with a certain probability, ensuring that the text objects are located in semantically coherent positions. The synthesised images adhere to perspective principles, avoiding potential depth estimation inaccuracies found in SynthText and VISD.

Figure 4 Examples of scene design (see online version for colours)



Domain randomisation theory aims to generate sufficiently varied synthetic data to bridge the gap with real data, allowing neural networks to treat real data as another variation (Tremblay et al., 2018). Based on the domain randomisation framework provided by Unity Perception (Borkman et al., 2021), several randomisers are defined to randomise various parameters in the scene. All randomisable parameter distributions follow uniform and truncated normal distributions (Burkardt, 2014). Examples of scene composition are shown in Figure 4. In this module, the following factors are considered for domain randomisation:

- *Transform:* An area is defined within the scene, where several coordinates spaced a certain distance apart are randomly generated using the Poisson disk algorithm. Random 3D models are placed at these generated coordinates, and parameters are defined to randomly adjust the transforms of these models.
- *Material texture:* A shader graph is designed to control the fragment shader of model materials. This shader graph receives diffuse maps, normal maps, and ambient occlusion maps as inputs, defining parameters to control the material's metallicity, smoothness, and diffuse colour. Several colour pairs $P = \{\{C_{b_1}, C_{t_1}\}, ..., \{C_{b_i}, C_{t_i}\}\}$ are collected from the background colour and text colour in real scene text images, where C_{b_i} and C_{t_i} are values in the HSV colour model. A certain amount of random variation is added to these values as the diffuse colours of background objects and text objects, ensuring their colour pairing is close to that of real scenes but not identical.
- *Ambient lighting:* HDRI maps are used as environment maps, which, along with additional directional lights and point lights, constitute the lighting of the scene. The environment map is randomly rotated around the scene's Y-axis, and the

intensity, hue offset, and positions of the directional lights and point lights can also be controlled by parameters.

- *Viewpoint:* A perception camera is setup in the scene, always facing the center of the scene. The parameters of the perception camera, such as position, field of view, and focal length, are then randomised. During the image synthesis process, the Unity Perception plugin captures the view observed by the perception camera and the corresponding segmentation map of the text instances.
- *JPG artefacts:* The output JPG images undergo random compression at varying ratios to introduce JPG artefacts, simulating possible image distortion found in real data.

3.3 Post-processing

This module outputs the final synthetic data, ensuring that the annotation format of the synthesised data is consistent with Total-Text to guarantee the general applicability of the data.

Figure 5 Post-processing steps, (a) text instance segmentation map (b) edge of the segmentation map (c) set of coordinates after removing internal vertices (d) Delaunay triangulation and removal of longer edges after triangulation (e) edge coordinates set after triangulation (f) final result after reducing vertices with Douglas Peucker algorithm (see online version for colours)

(a) (b) (c) (d) (e) (f)

The annotations for the synthetic data include transcriptions, orientation, and the text region. Since the synthesised text instances are in arbitrary orientations, the corresponding text regions are represented by polygonal coordinates $P = \{(x_{P_1}, y_{P_1}), (x_{P_2}, y_{P_2}), ..., (x_{P_n}, y_{P_n})\}$. The annotations for text transcriptions and orientations are derived from the information generated by the synthetic text models module. The polygonal bounding box annotations for the text areas are derived from the segmentation map $S = \{(x_{S_1}, y_{S_1}), (x_{S_2}, y_{S_2}), ..., (x_{S_n}, y_{S_n})\}$ of the text instances, which can be viewed as a polygon reconstruction problem. The edge of the segmentation map S is obtained by applying an edge detection algorithm. In this paper, we propose a new polygon reconstruction algorithm. Each column of the image is scanned to retain the coordinates of the maximum and minimum y values in the set of coordinates, and Delaunay triangulation is used to convert the point set V on the plane into a set of triangles T. A length threshold t is set to remove edges longer than t from the triangle set, yielding a contour formed by the remaining edge set. For the set of coordinates on the contour $C = \{(x_{C_1}, y_{C_1}), (x_{C_2}, y_{C_2}), ..., (x_{C_n}, y_{C_n})\}$, the Douglas-Peucker algorithm is used, with an approximation accuracy parameter ε set, to approximate the contour with fewer coordinate points, resulting in a polygon point set $P = \{(x_{P_1}, y_{P_1}), (x_{P_2}, y_{P_2}), ..., (x_{P_n}, y_{P_n})\}$. Finally, the data is reorganised to achieve the annotation

format that describes a text instance area with polygonal coordinates, consistent with Total-Text. The process of the algorithm is shown in Algorithm 1, and illustrative examples of each step and the final result are shown in Figure 5.

Algorithm 1 Convert segmentation map into polygon vertex text boxes

Input: Text instance segmentation map $S = (x_{S_1}, y_{S_1}), (x_{S_2}, y_{S_2}), ..., (x_{S_n}, y_{S_n})$, edge length threshold t, maximum distance from contour to approximated contour ε
Output: Polygon $P = (x_{P_1}, y_{P_1}), (x_{P_2}, y_{P_2}), ..., (x_{P_n}, y_{P_n})$

Use edge detection algorithm to obtain the edge coordinates set E from the segmentation map
Scan each column of E, retaining the vertices in E with the maximum and minimum y values in each column to form the set V
Perform Delaunay triangulation on V to get the set of triangle edges T
Create an empty list L
for e in T **do**
    if e is shorter than threshold t **then**
        Add e to the list L
    end if
end for
Connect the vertices in L, find the edges of L, represented by the point set $C = (x_{C_1}, y_{C_1}), (x_{C_2}, y_{C_2}), ..., (x_{C_n}, y_{C_n})$
Use Douglas Peucker algorithm with ε to approximate C into a polygon with fewer vertices, obtaining the polygon point set $P = (x_{P_1}, y_{P_1}), (x_{P_2}, y_{P_2}), ..., (x_{P_n}, y_{P_n})$

4 Experiments

4.1 Settings

In the experiments, 3D text models were synthesised using Blender 3.5, and images were generated based on Unity 2021.3.11f1 and Perception Package 1.0.0-preview-1. The defined randomisers, along with the specified parameters and distributions, are shown in Table 1. The hardware environment for synthesising the data consisted of a laptop equipped with a 2.20 GHz Intel Core i9-13900 CPU, Nvidia GeForce GTX 4060 GPU, and 16 GB RAM. The synthesis speed for data images was approximately 0.2 seconds per image, with a resolution of 800 $\times$ 600. The framework achieved a good synthesis speed under this configuration. A total of 10,000 images containing curve-oriented text instances and their corresponding annotations were synthesised using our method.

The DBNet model used in the experiments is derived from the open-source implementation of MindOCR. In all training tasks, the backbone network is ResNet-18, pretrained on the ImageNet dataset. The optimiser used is SGDM with a momentum of 0.9, weight decay set to 0.0001, and a batch size of 20, training for 1200 epochs each time. All model training and evaluation were conducted on a workstation equipped with an Ascend 910 chip.

Table 1 Design and distribution of the randomiser

Category	*Randomiser*	*Parameters*	*Distribution*
Camera	Camera	Field of view	$TN(40, 80, 60, 100)$
		Position X	$U(0, 1) + TN(-3, 3, 0, 1)$
		Position Y	$U(1, 3) + TN(-3, 3, 0, 1)$
		Position Z	$U(10, 15) + TN(-3, 3, 0, 1)$
3D object	Background	Scale	$TN(0.8, 1.2, 1, 0.005625)$
		Rotation X	$TN(-15, 15, 0, 25)$
		Rotation Y	$TN(-45, 45, 0, 225)$
		Rotation Z	$TN(-45, 45, 0, 225)$
	Foreground	Scale	$U(0.7, 1.3)$
Light	Lights	Light intensity	$U(0, 1)$
		Position X	$U(-6, 6)$
		Position Y	$U(1, 5)$
		Position Z	$U(-7.5, 7.5)$
Material	Materials	Multiply	$TN(10, 30, 20, 25)$
		Metallic	$TN(0, 1, 0.8, 0.04)$
		Smoothness	$TN(0, 1, 0.2, 0.04)$
		Colour offset H	$TN(-0.2, 0.2, 0, 0.01)$
		Colour offset S	$TN(-0.2, 0.2, 0, 0.01)$
		Colour offset V	$TN(-0.2, 0.2, 0, 0.01)$
Skybox	Skybox	Rotation Y	$U(0, 360)$
Post-process	Volume	Vignette intensity	$U(0, 0.5)$
		Fixed exposure	$TN(11, 15, 13, 0.5625)$
		White balance temperature	$U(-5, 5)$
		Film grain intensity	$U(0, 1)$
		Lens distortion intensity	$U(-0.2, 0.2)$
		Contrast	$U(-30, 30)$
		Saturation	$U(-30, 30)$

Notes: $U(a, b)$ denotes a uniform distribution between minimum a and maximum b. $TN(a, b, \mu, \sigma^2)$ denotes a truncated normal distribution with minimum a, maximum b, mean μ, and variance σ^2.

4.2 Evaluation metrics

In the experiments, the Deteval protocol (Wolf and Jolion, 2006) was used as the evaluation protocol, with the F-measure as the quantitative indicator for performance assessment. The formula for calculating the F-measure is as follows:

$$F\text{-}measure = \frac{2PR}{P + R} \tag{1}$$

where P and R represent the precision and recall of the detection, respectively. The formulas for calculating P and R are given by:

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

During the evaluation process, the intersection over union (IOU) value between the predicted boxes and the annotated boxes in the labels is calculated to determine whether the detected targets are positive or negative samples. Detected targets with an IOU greater than the threshold are considered positive samples, while those with an IOU less than the threshold are regarded as negative samples. Here, TP denotes the number of positive samples correctly identified as positive, FP denotes the number of negative samples incorrectly identified as positive, and FN denotes the number of positive samples incorrectly identified as negative.

4.3 Datasets

4.3.1 Synthetic datasets

The following synthetic datasets were selected for comparative experiments to investigate the impact of different types of synthetic data on the performance of scene text detectors. To ensure the fairness of the comparisons, the number of synthetic images used for training was limited to 10,000 per experiment.

- *SynthText* (Gupta et al., 2016) is a large synthetic dataset containing a total of 858,750 images, generated by synthesising text on natural images. It is widely used for pre-training various scene text detectors. In the experiments, 10,000 images were randomly sampled for training.
- *VISD* (Zhan et al., 2018) is a synthetic dataset containing 10,000 images, each with a different background. The text regions exhibit diversity in font, size, colour, and arrangement.
- *SynthText3D* (Liao et al., 2020a) is a scene text dataset synthesised using Unreal and the UnrealCV plugin, containing a total of 10,000 images. In the experiments, images without text instances or with clearly erroneous annotations were excluded.

To explore the impact of various synthetic datasets as training sets on the performance of scene text detectors, we also randomly selected 5,000 images from the SynthBendText3D and VISD datasets to form a complementary dataset for the following experiments.

4.3.2 Real datasets

The training subsets of the following real datasets were selected in some experiments to explore the impact of real data on the performance of scene text detectors, while the test subsets will be used to evaluate the performance of the detectors.

- *Total-Text* (Ch'ng and Chan, 2017) is collected from real scenes and includes 1,255 training images and 300 test images. It contains various types of text, including horizontal, multi-directional, and curved text instances.
- *SCUT-CTW1500* (Yuliang et al., 2017) is also collected from real scenes, consisting of 1,000 training images and 500 test images, with curved text instances described using polygons with up to 14 coordinate points.

4.4 Qualitative analysis

For scene text detection models, the quality of images in the training set and the shape of text instances are decisive factors affecting the model's performance on the test set. Figure 6 presents a visual comparison between the synthesised data in this paper and instances from other synthetic datasets. This paper utilises currently popular methods for synthesising data based on 3D engines. Unlike SynthText and VISD, which embed text into 2D images, SynthBendText3D can simulate more complex and diverse environmental conditions, while also placing text instances in more reasonable locations. Moreover, the text instances in SynthBendText3D are oriented in arbitrary directions and described in the form of polygon coordinates, which provides greater diversity in text shapes compared to other synthetic datasets and results in more reasonable annotation formats.

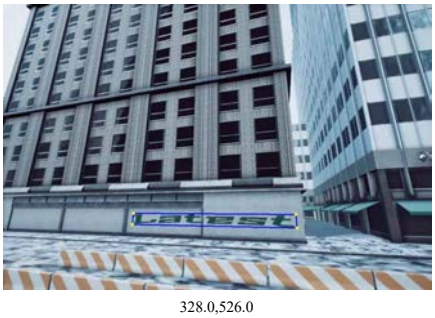
Figure 6 Examples of images from different synthetic scene text data and their annotation styles (see online version for colours)

array([413.37543, 523.0051 , 522.609 , 412.9793], dtype=float32)
array([32.585564, 34.57293 , 56.42428 , 54.436913], dtype=float32)
Lines:

(a) SynthText

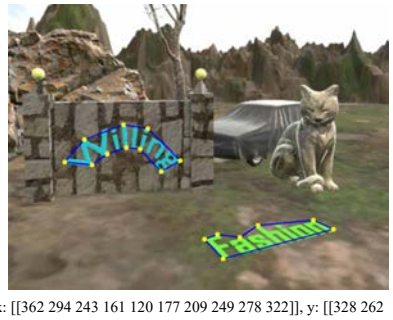
17,53,35,51,36,65,18,67,510
16,68,41,66,42,80,17,82,ECO
558,137,614,166,604,185,548,155,Timms
424,141,452,147,445,180,417,174,OFF
6,142,35,142,35,157,6,157,and
485,4,502,8,496,35,479,30,Our

(b) VISD



328.0,526.0
325.0,554.0
740.0,563.0
737.0,529.0
0

(c) SynthText3D



x: [[362 294 243 161 120 177 209 249 278 322]], y: [[328 262 254 285 331 337 314 302 305 351]], ornt: [u'c'], transcriptions: [u'Willing']
x: [[683 641 525 487 440 413 452]], y: [[472 453 464 489 483 493 534]], ornt: [u'm'], transcriptions: [u'Fashion']

(d) SynthBendText3D

Notes: Vertices represented by yellow coordinate points, text areas by blue lines.

4.5 Experiments results and evaluation

4.5.1 Training with synthetic data

In this section, experiments were conducted using the synthetic data from SynthBendText3D and three other types of synthetic data as training sets, while using the test subsets of Total-Text and SCUT-CTW1500 to evaluate performance. The test results on real data are shown in Table 2. When using Total-Text as the test set, the model trained with our method's synthetic data achieved F-measure values that were 6.82%, 0.61%, and 15.02% higher than those trained with SynthText, VISD, and SynthText3D, respectively. Mixing our method's synthetic data with an equal amount of VISD data for pre-training resulted in the best F-measure value for the scene text detector, demonstrating that the combination of different types of data can expand the training domain, thereby enhancing the robustness of the trained scene text detector. When using SCUT-CTW1500 as the test set, the model trained with our method's synthetic data achieved F-measure values that were 5.8%, 1.19%, and 8.28% higher than those trained with SynthText, VISD, and SynthText3D, respectively, with the mixed dataset yielding the best F-measure value.

Table 2 Results of training the scene text detector using different synthetic data

Training data	*Total-Text*			*SCUT-CTW1500*		
	P	*R*	*F*	*P*	*R*	*F*
SynthText 10K	72.36	50.70	59.62	66.97	46.49	55.86
VISD 10K	75.85	58.15	65.83	72.21	52.02	60.47
SynthText3D 10K	82.77	37.29	51.42	77.30	40.77	53.38
SynthBendText3D 10K	73.11	60.89	66.44	71.51	54.19	61.66
SynthBendText3D 5K + VISD 5K	80.18	59.01	*67.98*	75.85	53.30	*62.61*

Notes: In the table, 'P', 'R', and 'F' represent precision, recall, and F-measure, respectively. '5K' and '10K' indicate the number of data points in the dataset.

Visual results of the models trained with different datasets on the test set are shown in Figure 7. Other synthetic datasets do not include irregular text instances, which makes it difficult for the models to achieve good detection results on irregular text in the test set. In contrast, the model trained on SynthBendText3D performs well in detecting such text.

4.5.2 Fine-tuning in real data

Based on the training models obtained in the previous section, we continued to fine-tune the models using transfer learning, with the training subsets of Total-Text and SCUT-CTW1500 as the training sets. The test results on the test subsets of Total-Text and SCUT-CTW1500 are shown in Table 3. The F-measure of the models significantly improved, with the model pre-trained using our method's synthetic data mixed equally with VISD achieving the highest F-measure value.

Figure 7 Visualisation of detection results on the test set using models trained with different datasets, with detected text areas indicated by green boxes (see online version for colours)

(a) SynthText (b) VISD (c) SynthText3D (d) SynthBendText3D

Table 3 Results of fine-tuning the scene text detector using real data under the training model obtained in the previous section

Training data	*Total-Text*			*SCUT-CTW1500*		
	P	*R*	*F*	*P*	*R*	*F*
SynthText 10K	83.90	82.62	83.26	82.92	81.87	82.39
VISD 10K	85.23	85.03	83.60	81.48	85.33	83.36
SynthText3D 10K	80.29	83.20	81.72	85.42	80.51	82.89
SynthBendText3D 10K	83.47	84.68	84.07	82.63	85.96	84.26
SynthBendText3D 5K + VISD 5K	85.14	83.57	*84.35*	86.62	84.04	*85.31*

4.5.3 Combining synthetic and real data for training

In this section, we mixed different types of synthetic data with the training subsets of Total-Text and SCUT-CTW1500 to train the scene text detector. To evaluate the effectiveness of incorporating synthetic data, we also included a baseline experiment without synthetic data. The evaluation results of the models on the test subsets of Total-Text and SCUT-CTW1500 are shown in Table 4. The experiments demonstrate that the inclusion of various synthetic datasets can enhance the F-measure of the scene

text detector, validating the effectiveness of synthetic data. When using Total-Text as the test set, the F-measure values of models trained with our method's synthetic data were higher than those trained with SynthText, VISD, and SynthText3D by 1.36%, 0.83%, and 0.10%, respectively. Similarly, when using SCUT-CTW1500 as the test dataset, the F-measure values were higher than those of SynthText, VISD, and SynthText3D by 0.94%, 0.08%, and 1.14%, respectively. The model trained on the dataset obtained by mixing our method's synthetic data with VISD achieved the highest F-measure value.

Table 4 Results of training the scene text detector on a combination of synthetic and real data

Training data	*Total-Text*			*SCUT-CTW1500*		
	P	*R*	*F*	*P*	*R*	*F*
Real	84.18	81.90	83.02	79.64	80.89	80.26
SynthText 10K + Real	86.07	81.17	83.55	83.32	81.55	82.43
VISD 10K + Real	85.98	82.26	84.08	85.64	81.06	83.29
SynthText3D 10K + Real	87.78	82.03	84.81	84.07	80.46	82.23
SynthBendText3D 10K + Real	86.59	83.30	84.91	84.56	82.21	83.37
SynthBendText3D 5K + VISD 5K + Real	87.97	82.94	*85.38*	85.88	84.02	*84.94*

5 Conclusions and future work

In this paper, we proposed SynthBendText3D, a framework for synthesising scene text data in arbitrary orientations based on a 3D graphics engine. This framework synthesises a large number of text 3D models and places them at specific positions in a 3D scene, utilising domain randomisation techniques to randomise various parameters in the 3D scene, resulting in diverse synthetic data. Additionally, an algorithm for polygon reconstruction is integrated into the framework, converting segmentation annotations of arbitrary orientation text instances into polygon coordinate annotations. The effectiveness of the synthesised data was validated through comparative experiments with other synthetic datasets and comprehensive experiments involving real data. In the future, we will expand the application scope of our framework, such as synthesising multilingual text data, introducing more text styles, and applying the synthesised data from the framework in the field of scene text detection.

References

Abou Akar, C., Tekli, J., Khalil, J., Yaghi, A., Haddad, Y., Makhoul, A. and Kamradt, M. (2024) 'Sordi.AI: large-scale synthetic object recognition dataset generation for industries', *Multimedia Tools and Applications*, pp.1–42.

Borkman, S., Crespi, A., Dhakad, S., Ganguly, S., Hogins, J., Jhang, Y-C., Kamalzadeh, M., Li, B., Leal, S., Parisi, P. et al. (2021) *Unity Perception: Generate Synthetic Data for Computer Vision*, arXiv preprint arXiv:2107.04259.

Burkardt, J. (2014) *The Truncated Normal Distribution*, Vol. 1, No. 35, p.58, Department of Scientific Computing Website, Florida State University.

Ch'ng, C.K. and Chan, C.S. (2017) 'Total-Text: a comprehensive dataset for scene text detection and recognition', in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, Vol. 1, pp.935–942.

Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A. and Katam, H. (2019) *Blenderproc*, arXiv preprint arXiv:1911.01911.

Ebadi, S.E., Jhang, Y-C., Zook, A., Dhakad, S., Crespi, A., Parisi, P., Borkman, S., Hogins, J. and Ganguly, S. (2021) *PeopleSansPeople: A Synthetic Data Generator for Human-Centric Computer Vision*, arXiv preprint arXiv:2112.09290.

Gupta, A., Vedaldi, A. and Zisserman, A. (2016) 'Synthetic data for text localisation in natural images', iIn *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2315–2324.

Heindl, C., Brunner, L., Zambal, S. and Scharinger, J. (2021) 'Blendtorch: a real-time, adaptive domain randomization library', in *Pattern Recognition, ICPR International Workshops and Challenges: Virtual Event, Proceedings, Part IV*, Springer, 10–15 January, pp.538–551.

Herzog, F., Chen, J., Teepe, T., Gilg, J., Hörmann, S. and Rigoll, G. (2023) 'Synthehicle: multi-vehicle multi-camera tracking in virtual cities', in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp.1–11.

Janai, J., Güney, F., Behl, A., Geiger, A. et al. (2020) 'Computer vision for autonomous vehicles: problems, datasets and state-of-the-art', *Foundations and Trends® in Computer Graphics and Vision*, Vol. 12, Nos. 1–3, pp.1–308.

Kim, T.S., Shim, B., Peven, M., Qiu, W., Yuille, A. and Hager, G.D. (2022) 'Learning from synthetic vehicles', in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp.500–508.

Lee, H., Jeon, J., Lee, D., Park, C., Kim, J. and Lee, D. (2023) 'Game engine-driven synthetic data generation for computer vision-based safety monitoring of construction workers', *Automation in Construction*, Vol. 155, p.105060.

Liao, M., Shi, B., Bai, X., Wang, X. and Liu, W. (2017) 'Textboxes: a fast text detector with a single deep neural network', in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

Liao, M., Song, B., Long, S., He, M., Yao, C. and Bai, X. (2020a) 'Synthtext3D: synthesizing scene text images from 3D virtual worlds', *Science China Information Sciences*, Vol. 63, No. 2, p.120105.

Liao, M., Wan, Z., Yao, C., Chen, K. and Bai, X. (2020b) 'Real-time scene text detection with differentiable binarization', in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp.11474–11481.

Lu, F., McCaffrey, C.S. and Kuo, E.I. (2011) *Foreign Language Abbreviation Translation in an Instant Messaging System*, US Patent 7,890,525.

Martinez-Gonzalez, P., Oprea, S., Castro-Vargas, J.A., Garcia-Garcia, A., Orts-Escolano, S., Garcia-Rodriguez, J. and Vincze, M. (2021) 'Unrealrox+: an improved tool for acquiring synthetic data from virtual 3D environments', in *2021 International Joint Conference on Neural Networks*, IEEE, pp.1–8.

Mavrogiannis, C., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A. and Oh, J. (2023) 'Core challenges of social robot navigation: a survey', *ACM Transactions on Human-Robot Interaction*, Vol. 12, No. 3, pp.1–39.

Mousavi, M., Khanal, A. and Estrada, R. (2020) 'AI playground: unreal engine-based data ablation tool for deep learning', in *International Symposium on Visual Computing*, Springer, pp.518–532.

Mumuni, A., Mumuni, F. and Gerrar, N.K. (2024) 'A survey of synthetic data augmentation methods in machine vision', *Machine Intelligence Research*, Vol. 21, No. 5, pp.831–869.

Nikolenko, S.I. (2021) *Synthetic Data for Deep Learning*, Vol. 174, Springer, Cham.

Qiu, W., Zhong, F., Zhang, Y., Qiao, S., Xiao, Z., Kim, T.S. and Wang, Y. (2017) 'UnrealCV: virtual worlds for computer vision', in *Proceedings of the 25th ACM International Conference on Multimedia*, pp.1221–1224.

Toro, J.V., Bolin, L., Eriksson, J. and Wiberg, A. (2024) 'Towards digital representations for brownfield factories using synthetic data generation and 3D object detection', *Proceedings of the Design Society*, Vol. 4, pp.2297–2306.

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S. and Birchfield, S. (2018) 'Training deep networks with synthetic data: bridging the reality gap by domain randomization', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp.969–977.

Wolf, C. and Jolion, J-M. (2006) 'Object count/area graphs for the evaluation of object detection and segmentation algorithms', *International Journal of Document Analysis and Recognition*, Vol. 8, No. 4, pp.280–296.

Yuliang, L., Lianwen, J., Shuaitao, Z. and Sheng, Z. (2017) *Detecting Curve Text in the Wild: New Dataset and New Solution*, arXiv preprint arXiv:1712.02170.

Zhan, F., Lu, S. and Xue, C. (2018) 'Verisimilar image synthesis for accurate detection and recognition of texts in scenes', in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp.249–266.

Zhang, J., Fukuda, T. and Yabuki, N. (2022) 'Automatic generation of synthetic datasets from a city digital twin for use in the instance segmentation of building facades', *Journal of Computational Design and Engineering*, Vol. 9, No. 5, pp.1737–1755.

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W. and Liang, J. (2017) 'East: an efficient and accurate scene text detector', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.5551–5560.

Zhu, Y. and Du, J. (2018) 'Sliding line point regression for shape robust scene text detection', in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, pp.3735–3740.

Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L. and Zhang, W. (2021) 'Fourier contour embedding for arbitrary-shaped text detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.3123–3131.