Artif Intell Rev (2017) 48:299–329
DOI 10.1007/s10462-017-9569-z

Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives

Kunfeng Wang[1] · Chao Gou[1] · Nanning Zheng[2] · James M. Rehg[3] · Fei-Yue Wang[1,4]

Published online: 18 July 2017

Abstract In the study of image and vision computing, the generalization capability of an algorithm often determines whether it is able to work well in complex scenes. The goal of this review article is to survey the use of photorealistic image synthesis methods in addressing the problems of visual perception and understanding. Currently, the ACP Methodology comprising artificial systems, computational experiments, and parallel execution is playing an essential role in modeling and control of complex systems. This paper extends the ACP Methodology into the computer vision field, by proposing the concept and basic framework of Parallel Vision. In this paper, we first review previous works related to Parallel Vision, in terms of synthetic data generation and utilization. We detail the utility of synthetic data for feature analysis, object analysis, scene analysis, and other analyses. Then we propose the basic framework of Parallel Vision, which is composed of an ACP trilogy (artificial scenes, computational experiments, and parallel execution). We also present some in-depth thoughts and perspectives on Parallel Vision. This paper emphasizes the significance of synthetic data to vision system design and suggests a novel research methodology for perception and understanding of complex scenes.

Keywords Visual perception · Complex scenes · Parallel Vision · ACP Methodology · Computer graphics · Image synthesis

✉ Fei-Yue Wang
feiyue@gmail.com

1 The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

2 Institute of Artificial Intelligence and Robotics (IAIR), Xi'an Jiaotong University, Xi'an 710049, China

3 School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA

4 Research Center for Computational Experiments and Parallel Systems Technology, National University of Defense Technology, Changsha 410073, China

1 Introduction

In the field of computer vision, visual perception and understanding is an attractive technology to acquire information from video images (Wang and Yao 2015; Wang et al. 2012, 2016; Gou et al. 2016; Liu et al. 2016; Ramezani and Yaghmaee 2016; Gould et al. 2008; Farabet et al. 2013; Szeliski 2010). The information is mostly about target objects in the scene, including object location, identity, appearance, and behavior. Scene-level information such as semantic segmentation and scene understanding is also useful. Humans possess an innate ability to acquire information from video images, but for machines, it is surprisingly difficult to acquire the same information. For decades, many key topics [e.g., object detection and tracking (Wang and Yao 2015; Wang et al. 2016; Gou et al. 2016; Liu et al. 2016), activity recognition (Ramezani and Yaghmaee 2016; Wang et al. 2012), image segmentation (Gould et al. 2008; Farabet et al. 2013)] have been studied to further knowledge of visual perception and understanding. As a result, many relatively mature techniques have been used in a range of real-world applications, including optical character recognition, medical image processing, surveillance and traffic monitoring, and automotive safety (Szeliski 2010).

In recent years, the effectiveness of a vision algorithm in complex scenes has become a major concern for computer vision researchers (Wang and Yao 2015; Wang et al. 2012, 2016; Gou et al. 2016; Liu et al. 2016; Ramezani and Yaghmaee 2016; Datondji et al. 2016; Goyette et al. 2014). Let us take the urban traffic scene as an example. Many natural factors such as bad weather, strongly cast shadow, and low illumination at night often result in vague image details. In addition, the target objects have varied types, poses, appearance and behavior patterns, and are likely to be partially occluded by other objects in the field of view of the camera. Due to the combined effects of these factors, it is extremely difficult to design a robust vision algorithm. Many vision algorithms have not been sufficiently trained and evaluated with large-scale diversified datasets. Although these algorithms work well in simple controlled environments, they might easily fail when encountering complex challenging scenes (Wang and Yao 2015; Wang et al. 2012, 2016; Gou et al. 2016; Liu et al. 2016; Ramezani and Yaghmaee 2016; Datondji et al. 2016; Goyette et al. 2014).

Before the boom of deep learning, a common methodology adopted by traditional vision algorithms was to craft image features [e.g., Haar-like wavelet, SIFT (Scale-Invariant Feature Transform), HOG (Histograms of Oriented Gradients), and LBP (Local Binary Pattern)] by hand and then train pattern classifiers [e.g., SVM (Support Vector Machine), AdaBoost, and CRF (Conditional Random Field)] using certain labeled datasets. In this way, occasional satisfactory experimental results could be obtained, such as the DPM (Deformable Parts Model) object detector (Felzenszwalb et al. 2010) and the CRF-based image segmentation method (Gould et al. 2008). However, these algorithms usually rely on small-scale datasets (e.g., INRIA Person (http://pascal.inrialpes.fr/data/human/), Caltech Pedestrian (http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/), and KITTI (http://www.cvlibs.net/datasets/kitti/) datasets), with the number of samples from thousands to hundreds of thousands. Therefore, the datasets can cover only a small portion of the scene space. In contrast, the recently popular deep learning approach has strong feature representation abilities and is able to automatically learn hierarchical features with end-to-end training (Krizhevsky et al. 2012; LeCun et al. 2015; Ren et al. 2017; He et al. 2016). In a wide range of recent academic challenges (such as image classification, object detection, and image segmentation), deep-learning-based methods significantly outperform traditional methods, and the performance has been improving year by year. Deep learning relies on large-scale labeled datasets [such as ImageNet (http://www.image-net.org/), PASCAL VOC (http://host.robots.ox.ac.uk/

pascal/VOC/), and MS COCO (http://mscoco.org/)], which often contain millions of annotated samples or even more, thus being able to cover a much larger portion of complex scene space.

To design vision models that can generalize well given novel scenes, we should demand not only that the size of the labeled dataset be sufficiently large, but also the dataset has enough diversity. Although the sizes of some recent datasets like ImageNet are large, they do not satisfy the demand of enough diversity, so that they are still unable to cover complex scenes. This situation is caused by two major reasons. First, it requires massive manpower to collect large-scale diversified data from complex scenes. ImageNet [20] is compiled by collecting natural images from the Internet, but cyberspace and physical space are not equivalent, and moreover, the collection of the images depends on the human perception of the person selecting the images and constructing the dataset. This is bound to cause dataset bias (Torralba and Efros 2011; Model and Shamir 2015). Second, it is time-consuming and error-prone to annotate large-scale diversified data. Especially under adverse weather and low illumination conditions, the image details are so vague that it becomes quite difficult for a human annotator to make precise annotations by observing the images with the naked eye. Due to the faultiness of the labeled datasets, there is no guarantee that a vision model learned from such datasets will work well in real-world settings.

To address the difficulty in collecting and annotating large-scale diversified datasets, an alternative solution is to use artificial scenes that simulate complex real scenes and generate our desired synthetic datasets. With the great progress in computer graphics and virtual reality (Bainbridge 2007; Taylor et al. 2007; Butler et al. 2012; Wulff et al. 2012; Veeravasarapu et al. 2015a, b, 2016) in the past decade, it is now possible to construct high-quality photorealistic artificial scenes. With artificial scenes, various ingredients of real scenes can be simulated, including illumination periods (daytime, nighttime, dawn, and dusk), weather conditions (sunny, cloudy, rainy, snowy, foggy, windy, etc), object categories (pedestrians, vehicles, roads, buildings, vegetation, etc) and subcategories. In addition, all factors such as scene geometry, object appearance and motion, and camera configurations can be fully controlled by human users. Thereby, large-scale diversified image datasets can be synthesized, and their ground-truth annotations can be generated automatically, including object position, motion trajectory, semantic segmentation, depth, optical flow, and so on. In recent years, synthetic data have been utilized to design and evaluate a variety of computer vision algorithms (Taylor et al. 2007; Butler et al. 2012; Wulff et al. 2012; Veeravasarapu et al. 2015a, b, 2016).

In this review paper, we survey the use of photorealistic image synthesis methods in addressing the problems of visual perception and understanding, from the perspective of Parallel Vision (see Sect. 3). Building on top of both the real and artificial scenes, Parallel Vision is a virtual/real interactive vision computing methodology. It is inspired by the ACP (i.e., Artificial systems, Computational experiments, and Parallel execution) Methodology for modeling and control of complex systems (Wang 2004, 2010, 2013; Wang et al. 2016). For Parallel Vision, photorealistic artificial scenes are constructed to simulate the environmental conditions potentially occurring in real scenes, and precise ground-truth annotations are generated automatically. Combing large-scale synthetic data and a certain amount of real data, the machine learning and computer vision models can be trained more effectively. Based on artificial scenes, a range of computational experiments can be carried out to evaluate the performance of vision algorithms. If parallel execution is implemented by operating the vision model in both the real and artificial scenes concurrently, the model learning and evaluation can be implemented online and for a long term, so that the vision system is boosted continuously. As a result, the vision system is improved in effectiveness and adaptability. To sum up, Parallel Vision integrates several advanced technologies (such as computer graphics, virtual

reality, and machine learning) and will promote the research process of intelligent vision computing. It is worthwhile to note that the term "parallel" throughout this paper means the virtual/real interaction and parallel execution between artificial scenes and the real scene, rather than conventional two-camera stereo vision systems (Bertozzi and Broggi 1998).

This paper aims to emphasize the significance of synthetic data to vision system design and suggest a novel research methodology for perception and understanding of complex scenes. Specifically, the contributions of this paper are twofold. (1) We make a comprehensive review of previous works related to Parallel Vision, in terms of synthetic data generation and utilization. In particular, we detail the utility of synthetic data for feature analysis, object analysis, scene analysis, and other analyses. (2) We propose the basic framework of Parallel Vision as well as some in-depth perspectives. This ACP-based methodology is expected to promote the development of intelligent vision computing and speed up its industrialization.

The remainder of this paper is organized as follows. In Sect. 2, the previous works related to Parallel Vision are reviewed. The framework of Parallel Vision is detailed in Sect. 3, and some in-depth perspectives on Parallel Vision are presented in Sect. 4. Finally, a conclusion is drawn in Sect. 5.

2 Related works

As pointed out by Bainbridge (2007), computer-generated virtual worlds where virtual characters can interact in realistic manners have great potential as sites for research in the social, behavioral, and economic sciences, as well as in human-centered computer science. A variety of research methodologies that scientists have been exploring, including formal experimentation and quantitative analysis, can be used in virtual worlds. Nowadays, techniques associated with the construction of virtual worlds and artificial scenes are developing rapidly and playing an important role in scientific research. In this section, we review the related works in terms of synthetic data generation and utilization. The structure of these works is depicted as a schematic in Fig. 1.

2.1 Synthetic data generation

Based on the latest advances in computer graphics, some researchers are able to produce photorealistic synthetic data for vision system design and evaluation, as shown in Fig. 2. ObjectVideo Virtual Video (OVVV), presented by Taylor et al. (2007), is a publicly available visual surveillance simulation test bed based on the game engine Half-Life 2. The tool simulates multiple synchronized video streams from a variety of camera configurations, including static, PTZ, and omni-directional cameras, in a virtual environment populated with virtual humans and vehicles. To support performance evaluation, OVVV generates detailed automatic ground truth for each frame including target centroids, bounding boxes, and pixel-wise foreground segmentation. The authors described several realistic, controllable noise effects including pixel noise, video ghosting, and radial distortion to improve the realism of synthetic video and provide additional dimensions for performance testing. Several indoor and outdoor virtual environments developed by the authors were described to illustrate the range of possible testing scenarios using OVVV. Considering that ground truth optical flow is difficult to measure in real scenes with natural motion, Butler et al. (2012), Wulff et al. (2012) introduced an optical flow dataset derived from the open source 3D animated short film *Sintel*. In contrast to the previous Middlebury dataset, this new dataset has some important properties: long sequences, large motions, specular reflections, motion blur, defocus blur,

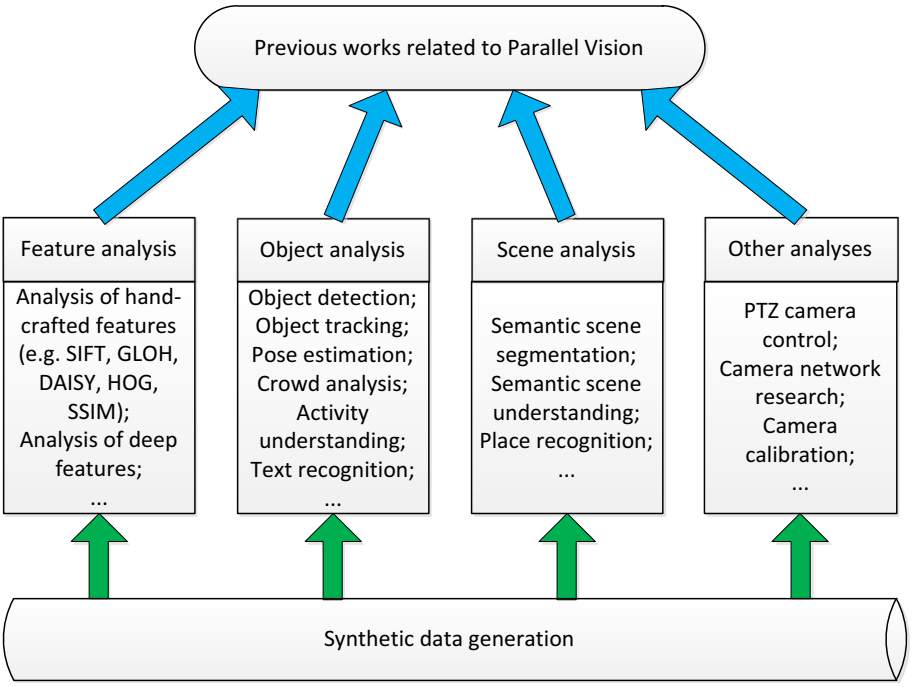

Fig. 1 Schematic of previous works related to Parallel Vision. Based on synthetic data, a variety of image/video analyses can be implemented

Fig. 2 Examples of synthetic images. *Top row* (*left* to *right*): OVVV data (Taylor et al. 2007), MPI-Sintel data (Butler et al. 2012; Wulff et al. 2012), synthetic data created in Veeravasarapu et al. (2015a, b, 2016), and SABS data (Brutzer et al. 2011). *Bottom row* (*left* to *right*): synthetic data created in Qureshi and Terzopoulos (2008), Virtual City data (Kaneva et al. 2011), SYNTHIA data (Ros et al. 2016), and GTA5 data (Richter et al. 2016)

and atmospheric effects. Since the graphics data that generate the movie is open source, it is possible to render scenes under conditions of varying complexity to evaluate where existing optical flow algorithms fail. To validate the utility of synthetic data, the authors compared the image- and flow-statistics of Sintel to those of real films and videos and showed they are similar. Recently, Veeravasarapu et al. (2015a, b, 2016) described a graphics simulation platform that is built on top of the open source graphics rendering tool *Blender*, and used that

platform for systematic performance characterization and tradeoff analysis for vision system design. They verified the utility of the platform in a case study of validating the rank-order consistency model in the contexts of global and local illumination changes, bad weather, and high-frequency noise. They also examined several invariance assumptions (order consistency, brightness constancy, gradient constancy, dichromatic scattering, and object shape feature) used for video surveillance settings and assessed the degree to which those assumptions deviate as a function of contextual variables on both graphics simulations and in real data. More recently, Qiu and Yuille (2016) presented an open-source tool called UnrealCV which can be plugged into the game engine Unreal Engine 4 (UE4) to help construct realistic virtual worlds from the resources of the game, virtual reality, and architecture visualization industries. These virtual worlds allowed them to access and modify the internal data structures enabling them to extract ground truth, control a virtual camera, and train and test computer vision algorithms.

Instead of synthesizing images purely from a virtual world, other researchers use some real images and 3D CAD models of an object class to generate a large number of synthetic images. For example, Rematas et al. (2014) proposed a technique to use the structural information extracted from a set of 3D models of an object class to improve novel-view synthesis for images showing unknown instances of this class. These novel views can be used to amplify training image collections that typically contain only a small number of views or lack certain classes of views entirely (e.g., top views). They extracted the correlation of position, normal, reflectance and appearance from computer-generated images of a few exemplars and used this information to infer new appearance for new instances. Rozantsev et al. (2015) proposed an approach to synthesizing images that are effective for training object detectors. Starting from a small set of real images, their algorithm estimates the rendering parameters required to synthesize similar images given a coarse 3D model of the target object. These parameters can then be reused to generate an unlimited number of training images of the object of interest in arbitrary 3D poses, which can then be used to increase classification performance. A key insight of their work is that the synthetically generated images should be similar to real images, not in terms of image quality, but rather in terms of features used during the detector training.

Procedural modeling deals with automatic or semi-automatic content generation by means of a program or procedure. Among many advantages, its data compression and the potential to generate a large variety of detailed content with reduced human intervention, have made procedural modeling attractive for creating virtual environments. Smelik et al. (2014) surveyed procedural methods that are useful for generating features of virtual worlds, including terrains, vegetation, rivers, roads, buildings, and entire cities. They focused particularly on the degree of intuitive control and of interactivity offered by each procedural method, because these properties are instrumental for designers. In order to support research of traffic simulation and human driving behavior, Prendinger et al. (2013) constructed a Virtual Living Lab using synthesis tools of OpenStreetMap, CityEngine, and Unity3D. They used an open source tool *OpenStreetMap* to create both the navigation network and the 3D road network, and used vehicle agents and the navigation segment agent to acquire knowledge of the surrounding environment. Recently, Dosovitskiy et al. (2017) proposed to train generative "up-convolutional" neural networks that are able to generate images of objects given object style, viewpoint, and color. They trained the networks on rendered 3D models of chairs, tables, and cars. Their experiments show that the networks do not merely learn all images by heart, but rather find a meaningful implicit representation of 3D models allowing them to assess the similarity of different models, interpolate between given views to generate the

missing ones, extrapolate views, and invent new objects not present in the training set by recombining training instances, or even two different object classes.

In general, artificial scenes are created using open source or commercial game engines and graphics simulation tools, rather than starting from scratch. Some commonly used tools include Unity3D, Half-Life 2, Mental Ray, OpenGL, Panda3D, Google 3D Warehouse, 3DS MAX, and Blender. For reading convenience, we summarize the existing (though not necessarily complete) publicly available synthetic datasets in Table 1, in terms of the dataset name, release year, simulation tool, and usage.

2.2 Feature analysis with synthetic data

Image features play an important role in computer vision, as they are used for a range of tasks such as stereo matching, panorama stitching, 3D reconstruction, and object, scene, and activity recognition. Because of their importance, analyzing and optimizing image features is a significant undertaking. In contrast to images captured from the real world, synthetic images from a virtual world allow complete knowledge of the scene geometry and full control of the environment, thus making it possible to study the impact of different factors of the environment on the image features in isolation (Kaneva et al. 2011; Pinto et al. 2011; Aubry and Russell 2015).

Kaneva et al. (2011) proposed to use a photorealistic virtual world to gain complete and repeatable control of the environment in order to evaluate image features. They used two sets of images rendered from the Virtual City (as shown in Fig. 3) and from the Statue of Liberty to evaluate the performance of a selection of commonly-used feature descriptors, including SIFT, GLOH (Gradient Location and Orientation Histogram), DAISY, HOG, and SSIM (the self-similarity descriptor). Their evaluation was conducted by matching keypoints. They found that the performance of the descriptors on similar datasets from the real world and virtual Statue of Liberty is similar and results in the same ranking of the descriptors. They then used the virtual world to study the effects on descriptor performance of controlled changes in illumination and camera viewpoint. They also studied the effect of augmenting the descriptors with depth information to improve performance. Pinto et al. (2011) compared five state-of-the-art visual features on invariant object recognition tasks, which include SIFT, PHOW (Pyramid Histogram Of visual Words), Pyramid HOG, Geometric Blur, and SLF (Sparse Localized Features). This comparison was conducted on a synthetic image set where the variation in object view was tightly controlled and the ground truth was known. The authors reported that most of the features perform poorly on invariant recognition, but that SLF consistently shows good performance in all of their tests.

More recently, Aubry and Russell (2015) proposed an approach to analyzing the variation of deep features generated by convolutional neural networks (CNNs) with respect to scene factors that occur in natural images. Such factors include intrinsic factors (object category, style, and color) and extrinsic factors (3D viewpoint and scene lighting configuration). Using a database of 3D CAD models, their approach analyzes CNN feature responses corresponding to different scene factors by fully controlling them via rendering. The rendered images are presented to a trained CNN and responses for different layers are studied with respect to the input scene factors. The authors observed important differences across the CNN layers for different scene factors. For example, the sensitivity to viewpoint decreases progressively in the last layers of the CNNs. They also demonstrated that their deep feature analysis based on computer-generated imagery is related to understanding the network representation of natural images.

Table 1 Some publically available synthetic datasets for computer vision research, which are ranked according to the release year

Dataset name	Release year	Simulation tool	Usage
OVVV (Taylor et al. 2007)	2007	Half-Life 2	Evaluating tracking and surveillance algorithms
Pedestrian dataset created in Marín et al. (2010), Vázquez (2013), Vázquez et al. (2014), Xu et al. (2014a, b)	2010	Half-Life 2	Learning pedestrian detection models
SABS (Brutzer et al. 2011)	2011	Mental Ray	Evaluating background subtraction algorithms
Virtual City and Statue of Liberty (Kaneva et al. 2011)	2011	3D CAD models and Mental Ray	Evaluating hand-crafted image features (e.g. SIFT, GLOH, DAISY, HOG, and SSIM)
MPI-Sintel (Butler et al. 2012; Wulff et al. 2012)	2012	Blender	Evaluating optical flow algorithms
BMC (Vacavant et al. 2013)	2012	SiVIC	Learning and evaluating background subtraction algorithms
AGORASET (Allain et al. 2012; Courty et al. 2014)	2012	Mental Ray	Evaluating low-level video crowd analysis methods, such as tracking or segmentation
Dataset created in Haltakov et al. (2013)	2013	VDrift	Learning and evaluating semantic segmentation algorithms
Dataset created in Sun and Saenko (2014), Sun et al. (2015), Peng et al. (2015)	2014	3D CAD models and 3DS MAX	Learning object detectors and evaluating the invariance of CNN features to low-level cues
Dataset created in Pinto et al. (2011)	2015	3D CAD models and OpenGL	Evaluating deep CNN features
SceneNet (Handa et al. 2015, 2016)	2015	CAD model repositories	Learning semantic segmentation models (indoor scenes)
SYNTHIA (Ros et al. 2016)	2016	Unity3D	Learning semantic segmentation models (street scenes)
GTA5 (Richter et al. 2016)	2016	Grand Theft Auto V and RenderDoc	Learning semantic segmentation models
Virtual KITTI (Gaidon et al. 2016)	2016	Unity3D	Learning and evaluating computer vision models for several video understanding tasks
RenderCar (Movshovitz-Attias et al. 2016)	2016	3D CAD models and 3DS MAX	Learning viewpoint estimation models

Table 1 continued

Dataset name	Release year	Simulation tool	Usage
LCrowdV (Cheung et al. 2016)	2016	Menge Engine for simulation and Unreal Engine for rendering	Learning pedestrian detection and crowd behavior models
CocoDoom (Mahendran et al. 2016)	2016	Doom game engine	Learning and evaluating computer vision methods
Dataset created in Chen et al. (2016)	2016	SCAPE and Blender	Learning human 3D pose estimation methods
FlyingThings3D, Monkaa, and Driving (Mayer et al. 2016)	2016	Blender	Learning and evaluating computer vision methods

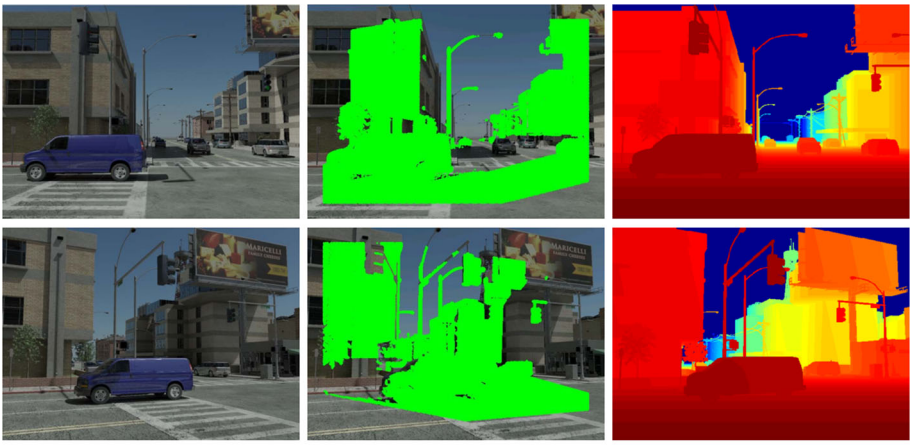

Fig. 3 Examples of synthetic images from the Virtual City (Kaneva et al. 2011). **a** Image pair of a scene under different viewpoints and illuminations. **b** The set of corresponding 3D points between the images in (**a**). **c** The corresponding depth maps of the images in (**a**)

2.3 Object analysis with synthetic data

Acquiring object-related information is an important motive of visual perception and understanding research. Object information is highly varied, e.g., positions, silhouettes, categories, identities, poses, moving directions and speeds, trajectories, and high-level activities. Due to the complexity of real-world scenes, it is laborious to collect and annotate large amounts of diversified real data. Instead, synthetic data generated with the latest computer graphics technology have been used in a range of object analysis tasks.

First, synthetic data have been proved useful for learning pedestrian and vehicle detectors. The works in Marín et al. (2010), Vázquez (2013), Vázquez et al. (2014), Xu et al. (2014a, b) are among the earliest attempts to learn pedestrian detectors from virtual-world datasets. The authors used the game engine Half-Life 2 to generate realistic virtual worlds and then explored whether virtual worlds can help learn appearance-based models for pedestrian detection in real-world images. Their experimental results concluded that virtual-world-based

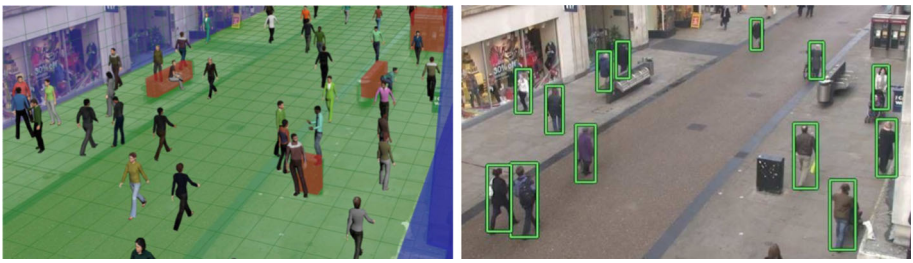

Fig. 4 Learning scene-specific pedestrian detectors without real data (Hattori et al. 2015). *Left*: geometrically consistent computer-generated training data. *Right*: scene-specific pedestrian detection results

training can provide excellent testing accuracy in the real world, but can also suffer from the dataset shift problem as real-world-based training does. Accordingly, they designed a domain adaptation framework, V-AYLA, in which they conducted active learning by collecting a few pedestrian samples from the target domain (real world) and then combining them with the many examples of the source domain (virtual world) in order to train a domain adapted pedestrian detector that operates in the target domain. V-AYLA reported the same detection accuracy as when training with many human-provided pedestrian annotations and testing with real-world images of the same domain. Recently, Hattori et al. (2015) studied the problem of designing a scene-specific pedestrian detector in a scenario where there are zero instances of real pedestrian data (i.e., without labeled or unlabeled real data during training), as shown in Fig. 4. The key idea of their work is to infer the potential appearance of pedestrians using geometric scene data and a customizable database of virtual simulations of pedestrian motion. They proposed an efficient discriminative learning method for generating a spatially-varying pedestrian appearance model that takes into account the perspective geometry of the scene. As a result, their method is able to learn a unique pedestrian classifier customized for every possible location in the scene. The experimental results showed that their approach outperforms generic pedestrian detection methods [i.e., HOG-SVM (Dalal and Triggs 2005) and DPM (Felzenszwalb et al. 2010)]. More surprisingly, their method using purely synthetic data is able to outperform models learned from real scene-specific data when data is limited. More recently, Johnson-Roberson et al. (2016) proposed a method to incorporate photorealistic computer images from a simulation engine to rapidly generate annotated data that can be used for training computer vision models. They demonstrated that the Faster R-CNN architecture, which is trained using only these synthetic annotations, performs better than the identical architecture trained on human annotated real-world data when tested on the KITTI dataset for vehicle detection. By training the detector in a data-rich virtual world, their work illustrated that real objects in real scenes can be learned and classified using solely synthetic data.

Second, some researchers try to create a large number of training images from freely available 3D CAD models, followed by learning generic object detectors with these images. For example, Sun and Saenko (2014), Sun et al. (2015), Peng et al. (2015) generated large-scale image datasets from 3D CAD models directly, and then used them to learn object detectors and explore the invariance of CNN features to low-level cues. They found that augmenting the training data of Deep CNN models with synthetic data can be effective, especially when real training data are limited or not well matched to the target domain. Most freely available CAD models capture 3D shape but are often missing other low-level cues, such as realistic object texture, pose, or background. In a detailed analysis, they used synthetic

CAD-rendered images to probe the ability of CNN to learn without these cues. Surprisingly, they found that when the CNN is fine-tuned on the target detection task, it exhibits a large degree of invariance to texture, color and pose, and less invariance to 3D shape. However, when pre-trained on generic ImageNet classification, the degree of invariance is lower and it learns better when the low-level cues are simulated. Pepik et al. (2015) used CAD models to generate three types of synthetic renderings: wire-frame, plain texture, and texture transfer. These renderings have different levels of photorealism. Their experimental results suggested synthetic data helps improve the overall quality of object detection and the improvement is directly proportional to photorealism.

Third, synthetic data can help learn and evaluate object tracking algorithms. For example, Gaidon et al. (2016) leveraged recent progress in computer graphics to generate fully labeled, dynamic, and photorealistic proxy virtual worlds. They proposed an efficient real-to-virtual world cloning method, and validated their approach by building and publicly releasing a new video dataset, called "Virtual KITTI", which was automatically labeled with accurate ground truth for object detection, tracking, scene and instance segmentation, depth, and optical flow, as shown in Fig. 5. They provided quantitative experimental evidence suggesting: (1) modern deep learning algorithms pre-trained on real data behave similarly in real and virtual worlds; (2) pre-training on virtual data improves multi-object tracking performance. As the gap between real and virtual worlds is small, virtual worlds enable measuring the impact of weather and imaging conditions on tracking performance, all other things being equal. They showed these factors may affect drastically otherwise high-performing deep models for tracking. In particular, bad weather (e.g., fog) causes the severest degradation of performance.

Fourth, object pose/viewpoint estimation also benefits from synthetic data, which is capable of supplying absolutely accurate pose/viewpoint annotations at negligible human cost. For example, Movshovitz-Attias et al. (2016) explored semi-automating dataset creation through the use of synthetic data and applied this method to object viewpoint estimation. Using state-of-the-art rendering software, they generated a large labeled dataset of cars rendered densely in viewpoint space. The generated dataset was used to train a deep convolutional network. They investigated the effect of rendering parameters on estimation performance and showed realism is important. They further reported that models trained on rendered data are as accurate as those trained on real images and that combining synthetic images with a small amount of real data improves estimation accuracy. Su et al. (2015) demonstrated that images rendered from 3D models can be used to train CNN for viewpoint estimation on real images. Their synthesis approach leveraged large 3D model collections to generate large-scale training data with fully annotated viewpoint information. Critically, they achieved this with negligible human effort, in contrast to previous efforts where training datasets have to be annotated manually. They showed that the CNN trained with rendered 3D model views can significantly outperform previous methods on the task of viewpoint estimation on 12 object classes from PASCAL 3D+. They also conducted extensive experiments to analyze the effect of synthesis parameters and input dataset scale on the estimation performance. Shotton et al. (2013) described two approaches to human pose estimation. Both can quickly and accurately predict the 3D positions of body joints from a single depth image without using any temporal information. The key to both approaches is the use of a large, realistic, and highly varied synthetic set of training images. This allows for learning models that are largely invariant to factors such as pose, body shape, field-of-view cropping, and clothing. Similarly, Danielsson and Aghazadeh (2014) studied the problem of estimating the pose of humans using pure RGB image input. Specifically, they used a random forest classifier to classify pixels into joint-based body part categories. Since the random forest requires a large number of training samples, they used computer-graphics-generated, synthetic training data. They proposed a

Fig. 5 The Virtual KITTI dataset (Gaidon et al. 2016). *Top*: a frame of a video from the KITTI multi-object tracking benchmark. *Middle*: the corresponding synthetic frame from the Virtual KITTI dataset with automatic tracking ground truth bounding boxes. *Bottom*: automatically generated ground truth for optical flow (*left*), semantic segmentation (*middle*), and depth (*right*)

new objective function for random forest training that uses the weakly labeled data from the target domain to encourage the learner to select features that generalize from the synthetic source domain to the real target domain. Recently, Chen et al. (2016) extended Su et al.'s work (Su et al. 2015) for human 3D pose estimation with a focus on domain adaptation and texture transfer. They found that pose space coverage and texture diversity are the key ingredients for the effectiveness of synthetic training data, and thereby presented a fully automatic, scalable approach that samples the human pose space for guiding the synthesis procedure and extracts clothing textures from real images. They further explored domain adaptation for bridging the gap between synthetic training images and real testing photos. Their experimental results demonstrated that CNNs trained with their synthetic images outperform those trained with real photos on 3D pose estimation tasks.

Fifth, synthetic data offers the possibility of generating a range of annotations for crowd video datasets and analyzing the crowded scenes. For example, Allain et al. (2012), Courty et al. (2014) presented a simulation-based crowd video dataset for evaluation of low-level video crowd analysis methods, such as tracking or segmentation. This dataset, named AGORASET, was synthesized by using the social force model to simulate and Mental Ray to

Fig. 6 A sample image and the ground-truth labels of virtual pedestrians (Cheung et al. 2016). *Left*: a sample image generated using LCrowdV and consisting of 858 pedestrian agents. *Right*: automatically generated labels of the pedestrians. It would be labor-intensive and error-prone if one labels every pedestrian by hand in such a crowded scene

render. Eight scenarios were designed to reflect eight typical crowd flow patterns observed in real-world situations. The associated ground-truth and metrics were also described, together with a case study of using AGORASET to evaluate an object tracker. Their synthetic dataset made it possible to evaluate crowd analysis methods, through full control of every factor of the synthetic video and automatic generation of the ground truth. Unfortunately, the photorealism of AGORASET is very limited and needs further improvement. Cheung et al. (2016) presented a procedural framework (named LCrowdV) to generate an arbitrary number of labeled crowd videos. The resulting crowd video dataset is used to learn and train computer vision models for crowded scene understanding. Their approach is composed of two components: a procedural simulation component for generating crowd movements and behaviors, and a procedural rendering component to generate synthetic videos or images. Each video or image is automatically labeled in terms of the environment, number of pedestrians, density, behavior, flow, lighting conditions, viewpoint, noise, etc. A sample image and the ground-truth labels of pedestrians are illustrated in Fig. 6. By combining LCrowdV with several real-world crowd datasets as the training dataset, they demonstrated that LCrowdV has the benefit to improve accuracy of pedestrian detection and crowd behavior classification algorithms.

In addition to the aforementioned topics, realistic synthetic data is also valuable to the study of other object analysis topics such as background subtraction (Brutzer et al. 2011; Vacavant et al. 2013; Sobral and Vacavant 2014), action/activity recognition and understanding (Ramezani and Yaghmaee 2016; Karamouzas and Overmars 2012; Fernández et al. 2011) text recognition (Jaderberg et al. 2014, 2016; Gupta et al. 2016; Ren et al. 2016), and biometrics (Cappelli 2015; Zuo et al. 2007; Ferrer et al. 2015; Charalambous and Bharath 2016; Galbally et al. 2012a, b; Correa et al. 2016; Luo et al. 2016). Interested readers are referred to the cited papers for more details.

2.4 Scene analysis with synthetic data

Compared to feature analysis and object analysis, scene analysis puts more emphases on scene-related information. In other words, scene analysis is a more holistic vision task. Syn-

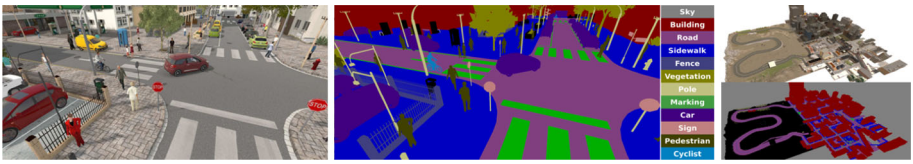

Fig. 7 The SYNTHIA dataset (Ros et al. 2016). A sample frame (*left*) with its semantic labels (*middle*) and a general view of the virtual city (*right*)

thetic data have been proved useful for scene analysis, including semantic scene segmentation (Veeravasarapu et al. 2016; Ros et al. 2016; Richter et al. 2016; Handa et al. 2015, 2016; Shafaei et al. 2016), semantic scene understanding (Zitnick et al. 2016; Chen et al. 2015), and place recognition (Sizikova et al. 2016).

First, synthetic data have begun to play a role in semantic scene segmentation, due to the inherent difficulty in labeling pixel-level semantic annotations manually and the photorealism of recently constructed artificial scenes. For example, Handa et al. (2015, 2016) presented an effective solution to the problem of indoor scene understanding—deep network trained with large numbers of rendered synthetic depth frames is able to achieve near state-of-the-art performance on per-pixel image labeling despite using only depth data. Their experiments showed that adding synthetic data improves the performance on NYUv2 and SUN RGB-D datasets for depth-only experiments and offers a promising route for further improvements in the state-of-the-art. Almost concurrently, Ros et al. (2016) presented SYNTHIA, a new dataset for semantic segmentation of driving scenes with more than 213,400 synthetic images, including both random snapshots and video sequences in a computer-generated virtual city. Simulating different seasons, weather, and illumination conditions from multiple viewpoints generates the images. Frames include pixel-level semantic annotations, as shown in Fig. 7. They used SYNTHIA in combination with publicly available real-world urban images with manually provided annotations. Their experiments with CNNs showed that the inclusion of SYNTHIA in the training stage significantly improves performance on the semantic segmentation task. Richter et al. (2016) presented an approach to rapidly creating pixel-accurate semantic label maps for images extracted from computer games. Although source code and internal operation of the commercial games they used are inaccessible, they managed to produce dense pixel-level semantic annotations for 25,000 images extracted from the game Grand Theft Auto V. The labeling process was completed in only 49 h, which is in contrast to at least 12 person-years if using conventional manual labeling. Their experiments showed that synthetic data is able to increase the performance of semantic segmentation models on real-world images and reduce the need for expensive hand-labeling. In addition to Ros et al. (2016), Richter et al. (2016), Handa et al. (2015, 2016), some other works (Veeravasarapu et al. 2016; Shafaei et al. 2016) on semantic segmentation got similar findings. In Haltakov et al. (2013), proposed another framework to generate a synthetic dataset for the task of multi-class image segmentation. They used synthetic data to train a CRF model and then analyzed the effects of using various combinations of features on the segmentation performance.

Second, synthetic data offers an excellent research platform for semantic scene understanding, due to its great ability to fully control the content in the scenario and automatically generate the desired labels. For example, Zitnick et al. (2016) proposed to study semantic information in abstract images created from collections of clip art. Abstract images provide several advantages over real images. They allow for the direct study of high-level semantics since they remove the reliance on noisy low-level object, attribute and relation detectors, or the tedious hand-labeling of real images. More importantly, abstract images allow the ability

to generate huge datasets of semantically similar scenes that would be nearly impossible with real images. The authors created 1002 sets of 10 semantically similar abstract images with corresponding written descriptions. They thoroughly analyzed this dataset to discover semantically important features, the relations of words to visual features and methods for measuring semantic similarity. Furthermore, they studied the relation between the saliency and memorability of objects and their semantic importance. Another interesting example relates to direct perception for autonomous driving, in which Chen et al. (2015) proposed to learn a mapping from an image to several meaningful affordance indicators of the road situation, including the angle of the car relative to the road, the distance to the lane markings, and the distance to cars in the current and adjacent lanes. This representation provides a set of compact yet complete descriptions of the scene to enable a simple controller to drive autonomously. To demonstrate this, they used a car racing video game TORCS for model training and testing. Specifically, they trained a deep CNN model using the recording from 12 h of human driving in TORCS and showed that their trained model can work well to drive a car in a diverse set of virtual environments. Testing their TORCS-based model on car-mounted smartphone videos further demonstrated good generalization ability in real-world perception.

Moreover, synthetic data is also beneficial to exploring other scene analysis tasks. One such example is visual place recognition, which refers to detecting when two images in a video sequence depict the same location, possibly under camera viewpoint or illumination-related appearance changes (Lowry et al. 2016). It is an important tool for robots to localize themselves in their surroundings. Sizikova et al. (2016) proposed to augment the place recognition process with individual separate intensity and depth networks trained on synthetic data. They showed that synthetic depth can be employed to train view-invariant CNNs that are useful for place recognition. Their experiments further indicated that combining descriptors from depth and intensity images shows improvement over intensity-only based place recognition, even when only a handful of aligned RGB-D frames are available for training.

2.5 Other analyses with synthetic data

In addition to the aforementioned feature-, object-, and scene-related analysis works, synthetic data have also been used to conduct other analyses, e.g., camera network research (Qureshi and Terzopoulos 2008; Starzyk and Qureshi 2013) and camera calibration (Creusot and Courty 2013).

As an alternative to real scenes, the computer-generated artificial scenes can be used as virtual experimental platform to conduct camera network research. For example, Qureshi and Terzopoulos (2008) proposed to study smart camera networks in virtual reality. A unique centerpiece of their work is the combination of computer graphics, artificial life, and computer vision technologies to develop such networks and experiment with them. Specifically, they demonstrated a smart camera network comprising static and active virtual surveillance cameras that provides extensive coverage of a large virtual public space, a train station populated by autonomously self-animating virtual pedestrians, as shown in Fig. 8. The simulated network of smart cameras performs persistent visual surveillance of individual pedestrians with minimal human intervention. On top of pedestrian detection and tracking, their camera control strategy addresses camera aggregation and handoff in the absence of camera calibration, detailed world model, or central controller. However, their simulated camera network has a limited scale (up to 16 virtual cameras and 100 virtual pedestrians) and the photorealism is low, lacking realistic imaging details such as shadow, highlight, or complex illumination. Afterwards, Starzyk and Qureshi (2013) presented a distributed virtual vision simulator that

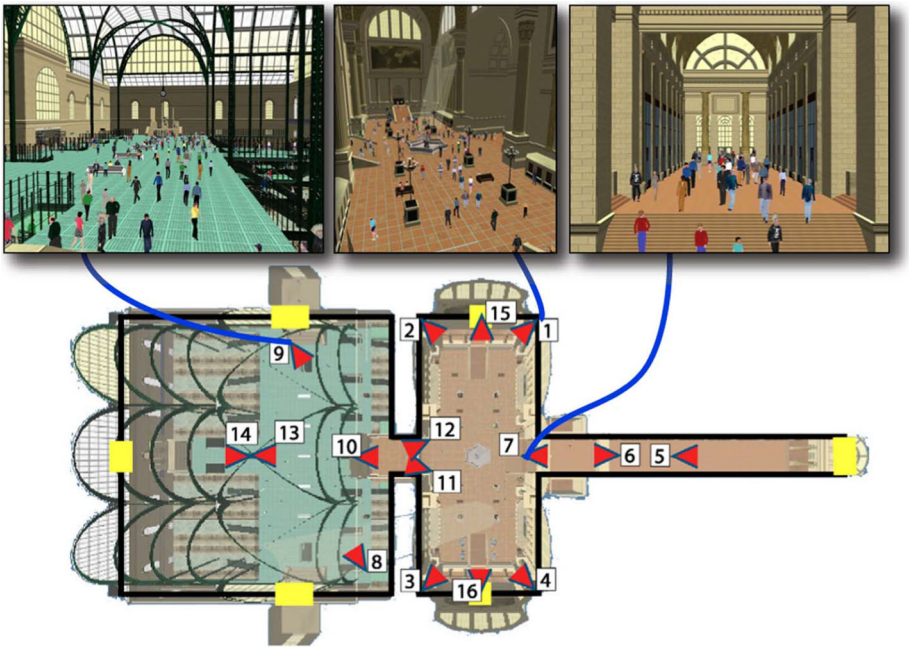

Fig. 8 Plan view of the virtual train station (Qureshi and Terzopoulos 2008), revealing the concourses and train tracks (*left*), the main waiting room (*middle*), and the shopping arcade (*right*). The *yellow rectangles* indicate pedestrian portals. An example camera network comprising 16 virtual cameras is illustrated. (Color figure online)

is capable of simulating large-scale camera networks. Simulated cameras generate synthetic video feeds that are fed into a vision processing pipeline to perform pedestrian detection and tracking. The visual analysis results are then used for subsequent processing, such as camera control, coordination, and handoff. Their virtual vision simulator is realized as a collection of modules that communicate with each other over the network. In consequence, they could deploy their simulator over a network of computers to simulate much larger camera networks and much more complex scenes. Specifically, their simulated camera network could comprise more than one hundred active PTZ and passive wide field-of-view cameras.

Another example explores the applicability of synthetic data to camera calibration. Creusot and Courty (2013) investigated the use of synthetic 3D scenes to generate temporally dense and precise segmentation ground truth of pedestrians in 2D crowd video data. They compiled a dataset of 1.8 million pedestrian silhouettes presenting human-to-human occlusion patterns that are likely to be seen in real crowd videos. They further discussed how to determine the camera tilt angle using the shape of the people in the video. Looking at a single frame only, their method is able to retrieve the camera tilt angle for angle views between 10° and 80° vertical angle, with the error less than 7° for low angle view and less than 3° for high tilt angles. This study demonstrated the utility of their proposed ground-truth data.

2.6 Summary

In summary, the previous works related to Parallel Vision take on three major trends. First, open-source and commercial simulation tools with increasingly stronger power emerge abundantly in recent years, making the constructed artificial scenes more and more photorealistic.

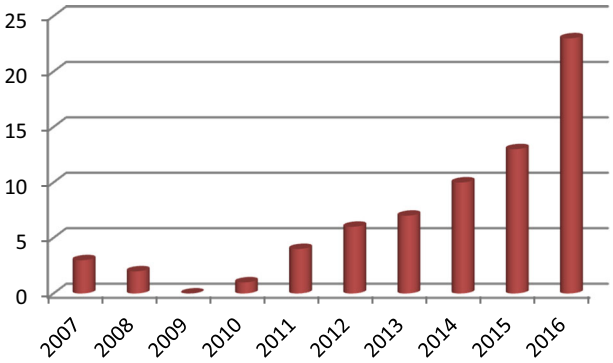

Fig. 9 Yearly number of the related works cited in Sect. 2

Figures 5 and 7 show two instances of photorealistic artificial scenes released in 2016. Second, the construction and utilization of artificial scenes have infiltrated into a wide range of computer vision tasks. From low-level tasks such as feature description and object detection, to mid-level tasks such as object tracking, and finally to high-level tasks such as action/activity recognition and understanding, virtual reality and artificial scenes have begun to play an important role. Third, more and more research papers related to Parallel Vision are published in international journals and conferences. Figure 9 shows the number of related works that are cited in this section, w.r.t. the publication years between 2007 and 2016. Although some works may be omitted, Fig. 9 clearly indicates that increasingly more related papers have been published in recent years. Moreover, in October 2016, the First Workshop on Virtual/Augmented Reality for Visual Artificial Intelligence was held in conjunction with the ECCV 2016, which is a clear signal that the role of computer graphics and virtual reality has received high attention in the computer vision community. Under these trends, we propose the concept and framework of Parallel Vision as well as some critical perspectives on it, in order to bring inspiration to computer vision researchers and promote the development of this fascinating research field.

3 Framework of Parallel Vision

3.1 Extending ACP into computer vision

In 2004, Fei-Yue Wang proposed the ACP Methodology for modeling and control of complex systems (Wang 2004), which can be expressed as:

$$\text{ACP} = \text{Artificial systems} + \text{Computational experiments} + \text{Parallel execution}.$$

Through this combination, ACP builds artificial cyberspace as the other half space, and then unites it with natural physical space into the complete "complex space" for solving complex problems. The emerging technologies of Internet of Things, Cloud Computing, and Big Data are ACP's core supporters. In essence, the fundamental idea of ACP is to build artificial systems as complex systems' virtual proxy, on top of which it is possible to conduct quantitative computational experiments and solve the complex problems more effectively. Fei-Yue Wang further proposed the concept of Parallel System through parallel execution between the unique real system and one or multiple artificial systems. After more than a

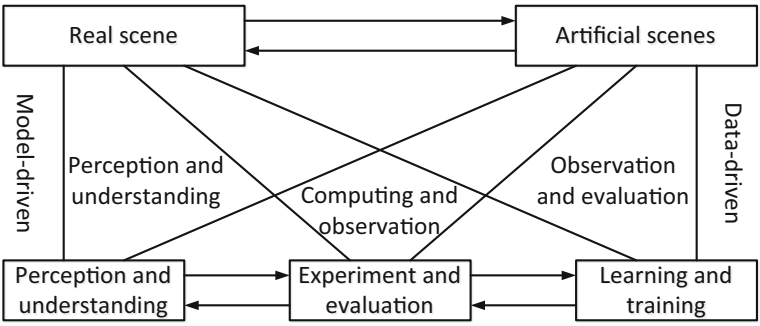

Fig. 10 Basic framework and architecture for Parallel Vision

decade of development, the ACP and Parallel System theories have gained applications in many fields, such as urban traffic management and control, Ethylene production process management, and social computing (Wang 2010, 2013; Wang et al. 2016).

In this paper, by extending ACP and Parallel System theories into the computer vision field, we present a novel vision computing methodology, named Parallel Vision. Our purpose is to develop a systematic approach to tackle the long-standing difficulty in visual perception and understanding of complex scenes. Figure 10 depicts the basic framework and architecture of Parallel Vision. In brief, Parallel Vision is composed of an ACP trilogy, which is described in the following subsections.

3.2 Artificial scenes

The first stage of Parallel Vision is to construct photorealistic artificial scenes by simulating a diverse variety of environmental conditions that may occur in real scenes, and accordingly to synthesize large-scale diversified datasets with precise annotations generated automatically. Generally speaking, the construction of artificial scenes can be regarded as "video game design", i.e., using the computer animation-like techniques to model the artificial scenes. The main technologies used in this stage include computer graphics, virtual reality, and micro-simulation. Computer graphics and computer vision, on the whole, can be thought of as a pair of forward and inverse problems. The goal of computer graphics is to synthesize image measurements given the description of world parameters according to physics-based image formation principles (forward inference), while the focus of computer vision is to map the pixel measurements to 3D scene parameters and semantics (inverse inference) (Veeravasarapu et al. 2015b). Apparently their goals are opposite, but can converge to a common point, i.e., Parallel Vision.

In many situations, due to the difficulty in data collection and annotation, we are unable to obtain a satisfying training dataset from real scenes. This situation is bound to hinder the design and evaluation of computer vision algorithms. Fortunately, the synthetic dataset compiled from artificial scenes can function as an alternative to real dataset. First, by means of the off-the-shelf computing resources, artificial scenes are ready to generate an infinite amount of training data, and by flexibly configuring the ingredients (e.g., scene layout, illumination, weather, and camera viewpoint) of artificial scenes, we can synthesize images with sufficient diversity. In consequence, it becomes possible to meet the requirement for large-scale diversified datasets. Second, while the real scene is usually unrepeatable, computer-generated artificial scenes can be repeated easily. By fixing some physical models and parameters while

tuning the others for the artificial scenes, we can customize the image formation factors. This allows us to evaluate our vision algorithms from various angles. Last but not least, there are some peculiar real scenes from which it is impossible to capture valuable training images, while the artificial scenes are seldom limited. Assuming we are developing a visual surveillance system for a novel battlefield, we may not be able to obtain video images with access to the enemy activities in advance. Under this situation, we are still able to construct artificial scenes in which virtual enemy troops are on the move, thereby using the synthetic images to develop the visual surveillance system. To sum up, constructing artificial scenes is of great significance, given that it offers a reliable data source (as a supplement to conventional real-scene data) for design and evaluation of vision systems.

3.3 Computational experiments

The second stage of Parallel Vision is to conduct various computational experiments with the artificial scene dataset as well as the available real dataset, in order for us to design the vision algorithms and evaluate them under complex environments. The main technologies used in this stage include machine learning, domain adaptation, and performance characterization. Most of the existing vision systems have not been validated thoroughly in complex environments, but rather been designed and evaluated under only a few common types of environments. As a result, it is quite uncertain whether such systems are valid or not when working under complex challenging conditions. To make a vision system truly effective, we should use artificial scenes to conduct thorough experiments. Specifically, we should establish a software laboratory for computer vision research, conduct computational experiments with the computer-generated artificial scenes, and design and evaluate vision algorithms thoroughly. In contrast to real scene-based experiments, the artificial scene-based experiments are fully controllable, observable and repeatable, and can really produce "big data", allowing for subsequent algorithm optimization.

The computational experiments have two operating modes: "learning and training" and "experiment and evaluation". The first mode "learning and training" focuses on the design of vision algorithms. In general, machine learning is located at the core of computer vision algorithms. Both classic statistical learning methods (e.g., SVM and AdaBoost) and the currently fashionable deep learning methods depend on "learning from data", which means training data plays a critical role in computer vision. If we combine a large-scale synthetic dataset with a certain scale of real data during training of machine learning models, the performance of vision algorithms can be improved to some degree. In particular, deep learning is a data-hungry method. The performance of deep learning models usually improves, as the amount of training data increases (Ren et al. 2017; Johnson-Roberson et al. 2016; Chen et al. 2015; Jones 2014). However, all machine learning methods suffer from a so-called "dataset bias" problem, that is to say, the training (source) domain has a distinct distribution with the testing (target) domain. Thus domain adaptation must be achieved. More perspectives on domain adaptation will be presented in Sect. 4.2. The second operating mode "experiment and evaluation" focuses on the evaluation of vision algorithms. It demands us to use the artificial scene dataset (as well as the available real dataset) to evaluate the performance of vision algorithms. Since we are able to completely control the generating factors (such as illumination, weather, objects, camera, etc) of artificial scenes, the validation of vision algorithms will be more thorough and more comprehensive. By means of performance characterization techniques (Thacker et al. 2008; Venetianer and Deng 2010), we can quantify the performance of vision algorithms w.r.t. environmental conditions. Such quantitative assessments will give us a confidence about the fielded efficacy of vision systems. To sum up, by

extending the computational experiments from real scenes to artificial scenes, we are able not only to expand the breadth of experiments but also to increase the depth of experiments, leading to improved performance of vision algorithms.

3.4 Parallel execution

The third stage of Parallel Vision is parallel execution, which means operating the vision model in both the real scene and one or multiple artificial scenes concurrently. In this way, the model learning and evaluation can be conducted online and for a long term. Through the virtual/real interaction between real-scene execution and artificial-scene execution, the vision system is optimized continuously. In addition to those technologies used in artificial scenes and computational experiments, online learning is particularly significant in this stage. From the related works elaborated in Sect. 2, it can be seen that many vision researchers have had ACP-like ideas, but mainly focused on A (Artificial scenes) and C (Computational experiments). In our opinion, all of the three stages of ACP are necessary to overcome the inherent difficulty in visual perception and understanding. Due to the unpredictable complexity, challenge, and variation of application environments, it is generally impossible (at least for now) to solve a vision computing problem once and for all. For that reason, parallel execution makes the virtual/real interaction a routine practice and attempts to boost the vision system at run time.

Parallel execution is driven by big data, and its primary characteristic is to construct artificial scenes in the loop. The available big data includes not only the real-time images, illumination and weather data from the physical space, but also large repositories of virtual object 3D models from the cyberspace. Analyzing the real-time images, one can (semi-)automatically acquire the scene/object geometry, illumination and weather conditions, etc. Combining them with the virtual object 3D models, it is possible to construct a variety of meaningful artificial scenes. Under the support of Internet of Things and Cloud Computing, a single real scene can correspond to multiple artificial scenes. As for constructing meaningful artificial scenes, we do not intend to clone or reconstruct the real scene, but intend to predict or foster the virtual proxy of real scenes. For an artificial scene to be meaningful, it must satisfy two demands: (1) it must follow and simulate the latest environmental challenges in the real scene, such as adverse illumination and weather conditions; (2) it must be beneficial to boosting the vision computing methods. On top of the online constructed artificial scenes, both of the operating modes of computational experiments, "learning and training" and "experiment and evaluation", can be applied to the vision system. As a result, the vision system is improved in its adaptability to the dynamic scene, and moreover it gets ready to meet future environmental challenges. To sum up, relying on the online construction and utilization of artificial scenes, parallel execution is a data-driven, virtual/real interactive approach to perception and understanding of complex scenes.

3.5 Summary

As a novel vision computing methodology, Parallel Vision makes an extension of the ACP and Parallel System theories (Wang 2004, 2010, 2013; Zhang et al. 2008; Zhu and Wang 2012; Yang and Wang 2007; Wang et al. 2016, 2017) for visual perception and understanding of complex scenes. For Parallel Vision, the ACP trilogy serves as a unity. Photorealistic artificial scenes are constructed to simulate and represent complex real scenes, computational experiments are utilized to design and evaluate a variety of vision

models, and parallel execution is finally implemented to optimize the vision system continuously.

4 Perspectives on Parallel Vision

In this section, we present some in-depth perspectives on the ACP trilogy of Parallel Vision, in order to shed some light and stimulate thought for the computer vision community.

4.1 Perspectives on artificial scenes

The first perspective on artificial scenes relates to the photorealism of computer-generated imagery, which is exactly a focus of attention in computer graphics. We do expect the synthetic images from artificial scenes to be as photorealistic as possible. In general, the photorealism depends on the level-of-detail in the domain models that simulate the real world as well as the level-of-fidelity of the rendering engine. To increase the level-of-detail, one needs to comply with physical rules when modeling a given application domain. For example, the 3D geometric structures and surface textures of artificial scenes and virtual objects should be consistent with those of real-world scenes and objects; when virtual pedestrians move in artificial scenes, they should be similar in speed and gait to real pedestrians; when virtual vehicles drive on the virtual road, they should comply with the driving behaviors of real vehicles. Nevertheless, all virtual models are simplifications of reality, leading to a trade-off as to what level of detail is included in the model. If too few details are included one runs the risk of getting very unrealistic images. If too much detail is included the model may become too complicated to handle. On the other hand, to maximize the level-of-fidelity of the rendering engine, one needs to use exact sampling and rendering processes from physics inspired models. This is mostly infeasible and computationally intractable. All in all, the photorealism of synthetic images is still limited, so that the utility and training value of a single synthetic image is generally lower than that of a single real image. This is a disadvantage of artificial scenes.

The second perspective on artificial scenes involves the diversity of labeled datasets. It is known that the real world is extremely diverse. In typical real scenes, there can be a variety of objects with distinct appearance and motion patterns. Taking an urban traffic scene as an example, the possible factors that constitute the scene are listed in Table 2. Even though the real world is diverse, due to the insuperable difficulty in data collection and annotation (see Sect. 1), real-scene datasets are often not diverse enough and suffer from the dataset bias issue (Torralba and Efros 2011; Model and Shamir 2015). By contrast, the computer-simulated artificial scenes are also extremely diverse, and can easily achieve data collection and annotation. Synthetic images from artificial scenes should still have bias, while injecting additional diversity. In accordance with the physical rules of real scenes, various scene layouts, objects, and even illumination and weather conditions can be simulated in artificial scenes. Moreover, we are able to completely control the generating factors of artificial scenes and synthesize our desired images. As a result, synthetic images seem more diverse than real image datasets compiled from the real world.

As another perspective, constructing artificial scenes with modern computers provides the flexibility to enlarge the scale of synthetic data. As long as the computing and storage resources are sufficient, the artificial scenes can be constructed to a size large enough. The synthetic images are captured using virtual cameras that can be easily configured in artificial scenes, without requiring the cumbersome installment of real cameras. Thus, the base number

Table 2 Possible factors that constitute an urban traffic scene

Factors	Contents
Static objects	Sky, building, road, pavement, fence, vegetation, lamp pole, lane marking, traffic sign, etc
Dynamic objects	Car, bus, truck, pedestrian, bicycle, motorcycle, etc
Seasons	Spring, Summer, Autumn, and Winter
Weather conditions	Sunny, cloudy, rainy, snowy, foggy, windy, etc
Light sources	Sun in daytime, street light and vehicle light at night, etc

of synthetic images can be much larger than the real image datasets. More importantly, since the artificial scenes are constructed in a bottom-up manner, a wide variety of ground truth annotations for the synthetic images can be generated automatically, including object position, silhouette, category, pose/viewpoint, motion trajectory, semantic scene segmentation, depth, optical flow, and so on. Having the ability to build large-scale labeled datasets is absolutely an advantage of artificial scenes.

In summary, artificial scenes have the potential to build synthetic image datasets with large scale and enough diversity, which helps to eliminate overfitting a computer vision model and enhance its generalization ability. However, the photorealism of current synthetic images is still limited, thus leading to dataset bias. Hence domain adaptation must be achieved to tackle this issue, as will be discussed in the next subsection. In spite of this, artificial scenes offer a powerful path forward as the ever-developing computer graphics technology can ultimately produce images that are indistinguishable from real-world images, yet still have all ground truth variables known and under parametric control.

4.2 Perspectives on computational experiments

Under the operating mode "learning and training", the synthetic images from artificial scenes are used to learn the computer vision model. Due to the limited photorealism of synthetic images, however, there is a gap (called dataset bias or domain shift) between the distributions of synthetic images and real images, so that the model learned from synthetic images cannot be directly applied to real images. To mitigate the effects of dataset bias, three domain adaptation strategies (Gopalan et al. 2015; Ganin et al. 2016; Ghifary 2016) could be used.

(1) Unsupervised domain adaptation—using a large amount of labeled synthetic data and a large amount of unlabeled real data together to train the model. Under this strategy, one does not have any labeled data from the target (real) domain. Some recent methods (Chen et al. 2016; Ganin et al. 2016; Ghifary 2016) propose to learn powerful features that are both discriminative for the label prediction task and invariant to the changes of domain, i.e., have very similar distributions in the source and the target domains.
(2) Supervised domain adaptation—using a large amount of labeled synthetic data and a small amount of labeled real data together to train the model. The unlabeled real data is not used in this strategy. One simple yet effective practice (Ros et al. 2016; Movshovitz-Attias et al. 2016) is to combine the labeled synthetic and real data as a common training set to train the model; another frequently adopted practice (Vázquez et al. 2014; Xu et al. 2014a, b; Handa et al. 2016; Gaidon et al. 2016) is first learning the model from labeled synthetic data and then fine-tuning it with labeled real data.

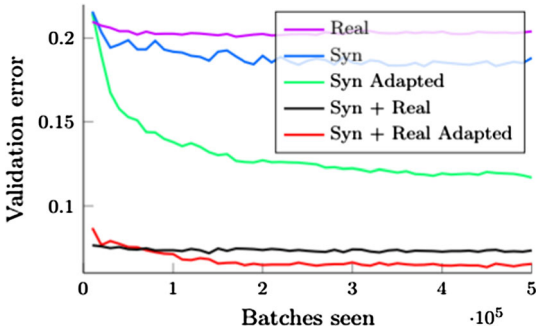

Fig. 11 Results for traffic sign classification with different model training strategies. *Syn* and *Real* denote available labeled data (100,000 synthetic and 430 real images respectively); *Adapted* means that ≈ 31,000 unlabeled target domain (real) images were used for domain adaptation. The best performance is achieved by employing the semi-supervised domain adaptation strategy. This example comes from Ganin et al. (2016). It is should be noted that after model training, the validation errors corresponding to the curves have the same order from *top* to *down* with the legend

(3) Semi-supervised domain adaptation — using a large amount of labeled synthetic data, a small amount of labeled real data, and a large amount of unlabeled real data together to train the model. Since the most data are utilized, this strategy is usually able to yield the best performance.

Let us quote an example from Ganin et al. (2016) to illustrate how useful the three domain adaptation strategies can be. This example relates to learning deep CNNs for classification of traffic signs (total 43 classes). Five model training strategies are evaluated, and Fig. 11 shows the change of validation error throughout the training. Because the number of labeled real images is too small (only 430 samples), the model learned purely from labeled real data results in the highest validation error. By learning purely from large amounts of labeled synthetic data (100,000 samples), the validation error can be reduced a little but is still high due to dataset bias. By using the unsupervised domain adaptation strategy, a sensible decrease in validation error is achieved although no labeled data from the target domain is utilized. Supervised domain adaptation further improves performance. However, the best performance is achieved by employing the semi-supervised domain adaptation strategy, perhaps because the most data are utilized. This example suggests that domain adaptation is beneficial to "learning and training" and that semi-supervised domain adaptation seems to be the most promising.

Under the operating mode "experiment and evaluation", the testing datasets from both artificial and real scenes are used sequentially to evaluate the vision algorithm. We suggest a circulation procedure for evaluating a vision algorithm. The algorithm should first be evaluated on synthetic dataset from artificial scenes. This is because artificial scenes are fully controllable and their ground truth can be automatically generated, so that we are able to freeze all scene-related factors (e.g., scene geometry, object appearance and motion, illumination, camera, etc) but adjust one (e.g., weather condition) during evaluation. This enables us to separate the impact of each factor on algorithm performance, evaluate the algorithm thoroughly, and discover the pros and cons of the algorithm. Even though a vision algorithm is confirmed effective in artificial scenes, it still needs to be evaluated on real dataset. This is because the vision algorithm will eventually be applied to real scenes, but there is a gap between artificial scenes and real scenes. If a vision algorithm performs badly in artificial or real scenes, it must be re-designed and re-evaluated. During "experiment and

evaluation", the hard examples could be automatically mined and added to the training set for algorithm re-design. This is followed by a new iteration, i.e., evaluate the vision algorithm first on artificial scenes and then on real scenes. And this iteration repeats until the algorithm passes the test in both kinds of scenes.

4.3 Perspectives on parallel execution

The aim of parallel execution is to online boost the vision system with a virtual/real interactive policy. Due to the unpredictable complexity, challenge, and variation of fielded environments, in many cases we do not have the ability to produce a perfect vision system once and for all. Parallel execution offers a more viable option, by first building an acceptable vision system at the beginning and then boosting it through online learning and optimization. For parallel execution, the artificial scenes are constructed and utilized in the loop. Under the support of Internet of Things and Cloud Computing, although the real scene is unique, multiple artificial scenes can interact in parallel with a single real scene. The artificial scenes are constructed by simulating the major generating factors (e.g., scene geometry, illumination, weather, etc) of the real scene. On the one hand, the artificial scenes should follow and simulate the dynamically changing environments in the real scene. For instance, the illumination and weather conditions in artificial scenes should be kept similar with those in the real scene. On the other hand, the artificial scenes could be made specific with respect to a real scene. This is particularly important for visual surveillance applications where the real scene is observed by a stationary camera, and the scene geometry is fixed. In general, a scene-specific vision model learned from scene-specific data performs better than a generic one (Hattori et al. 2015; Wang et al. 2014; Zeng et al. 2014). While collecting large amounts of training data from a specific real scene is time-consuming and laborious, constructing artificial scenes offers the feasibility of synthesizing large-scale diversified data for specific scenes and learning a scene-specific vision model. To sum up, the online constructed artificial scenes produce infinite labeled data that are essential to online learning, optimization, and evaluation of the vision system.

Both of the operating modes for computational experiments (i.e., "learning and training" and "experiment and evaluation") should be closely interwoven in this stage. For "learning and training", online bootstrapping (or hard example mining) can be implemented to automatically mine the hard examples where the vision model fails or performs badly (Felzenszwalb et al. 2010; Vázquez et al. 2014; Shrivastava et al. 2016; Loshchilov and Hutter 2016). The hard examples from both the online constructed artificial-scene data and the randomly selected real-scene data are accumulated to compose new batches of training data, which can then be used to fine-tune the vision model. The real-scene data may be offline or online, labeled or unlabeled. They are all useful to online learning, as domain adaptation (see Sect. 4.2) will be conducted to integrate information from both artificial and real scene data and suppress the dataset bias problem. Through such online learning and training, the vision model will adapt itself to the dynamically changing environments in the real scene. For "experiment and evaluation", the circulation procedure suggested in Sect. 4.2 can also be used to evaluate the vision model periodically. The online synthetic dataset from artificial scenes is first used for evaluation, and then the (offline or online) real dataset from real scenes is used. Performance characterization is conducted, and hard examples are automatically mined and accumulated. If a large drop in performance takes place, one needs to use new batches of training data to amend the vision model, or even substitute the previous model with a better-performing one. This procedure repeats continuously! As a result, the vision system is continuously optimized by parallel execution.

Fig. 12 Constructing artificial scenes by superimposing virtual pedestrians onto the real background images

4.4 Perspectives on applications

The Parallel Vision methodology can be applied to a broad range of vision computing tasks, including visual surveillance (Neves et al. 2016), robotic vision (Halim et al. 2016), medical imaging (Angel and Mary 2017), and so on. The ability of Parallel Vision to construct controllable artificial scenes and synthesize photorealistic images is rather useful for CVPR (computer vision and pattern recognition) research. By means of Parallel Vision, a CVPR system can learn from synthetic images and be evaluated in artificial scenes, which provide more flexibilities than actual images and scenes.

We have applied the Parallel Vision methodology to the study of visual surveillance. Given a specific surveillance scene, we can train object detectors even without any real training data. Since the camera is stationary and a background image can be estimated and updated online, we can construct an artificial scene to augment the real scene by superimposing virtual pedestrians onto the background image, as shown in Fig. 12. The virtual pedestrians moving in artificial scenes are exploited to simulate real pedestrians moving in actual scenes. At the same time, ground-truth annotations for the bounding boxes of virtual pedestrians are generated automatically. The synthetic images are collected to compose the training set, which can be used to train pedestrian detectors in case of lacking any real training data. Our experimental results indicate that training Faster R-CNN with scene-specific synthetic images increases the AP (average precision) of pedestrian detection by a large margin (averagely 15.8% in our experiment), compared to training Faster R-CNN with the generic PASCAL VOC dataset. From this case study of Parallel Vision, it can be expected that purely synthesized datasets are enough to train high-performance CVPR models.

The Parallel Vision methodology can also be applied to robotic vision research. As a kind of wheeled robot, on-road intelligent vehicle should perceive the surrounding situations and guide the vehicle to drive (Halim et al. 2016). Due to the complexity of real-world traffic scenes and the limitations of real datasets, it is difficult to design and evaluate the visual perception model based solely on real datasets and scenes. Artificial scenes are considered as a complementary space, in which model learning and evaluation can be conducted effectively and safely. We have constructed an artificial urban road network, in which virtual vehicles, pedestrians and cyclists move like real objects. The illumination and weather conditions can be controlled completely, simulating complex environmental conditions in the real world. Figure 13 shows the fidelity and diversity of synthetic images captured from our constructed artificial scenes. We are employing this artificial road network to develop new perception and decision methods for intelligent vehicles. Preliminary results have proved the feasibility of Parallel Vision.

Medical imaging is another field where Parallel Vision can play an important role. In this application field, CVPR systems should detect and segment various pathological tissues from radiology images (X-ray, ultrasound, computed tomography (CT), positron emission

Fig. 13 Virtual images synthesized from our constructed artificial scenes under different weather conditions (*Left top*: cloudy weather; *right top*: sunny weather; *left bottom*: foggy weather; *right bottom*: rainy weather)

tomography (PET) and magnetic resonance image (MRI)) and histopathology images (Angel and Mary 2017). In order to train the CVPR systems, a number of training images should be collected and the ground-truth annotations (including pathological tissue segmentation and categorization) should be labeled by human experts. However, the collectable medical images with pathological tissues are limited and the experts are always sparse. It is practically impossible to collect and annotate large-scale diversified medical images with various pathological tissues. As an alternative, synthetic images with virtual pathological tissues can be generated by using the virtual reality technology. Synthetic images and real images can be combined to design the medical CVPR systems. This should be a promising application of Parallel Vision in medical imaging.

5 Conclusion

In this article, we survey the use of photorealistic image synthesis methods in addressing the problems of visual perception and understanding, from the perspective of Parallel Vision. We extend the ACP Methodology into the computer vision field and propose the concept and basic framework of Parallel Vision. As a novel vision computing methodology, Parallel Vision builds on top of computer graphics and virtual reality to explore a systematic method for visual perception and understanding of complex scenes. Parallel Vision is composed of a new ACP trilogy. Photorealistic artificial scenes are constructed to simulate and represent complex real scenes, making it possible to collect and annotate large-scale diversified datasets. Then

computational experiments are utilized to learn and evaluate a range of vision models, leading to improved performance. Finally, parallel execution is implemented to optimize online the vision system, making virtual/real interaction a routine practice of vision computing.

The Parallel Vision related research has attracted much attention in the computer vision community. In recent top-class computer vision conferences and journals (e.g., ICCV, CVPR, ECCV, IJCV, and PAMI), a great many works that use computer graphics to help address a range of vision computing problems have been published. With the rapid development of computer graphics, the computer-generated artificial scenes will be more photorealistic in the future, thus providing a more reliable foundation support for Parallel Vision. We believe that Parallel Vision will become an important research branch in the computer vision field. In particular, the combination of Parallel Vision and Deep Learning has great potential to promote the development of intelligent vision systems and speed up the process of industrialization.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 61533019 and No. 71232006).

References

Allain P, Courty N, Corpetti T (2012) AGORASET: a dataset for crowd video analysis. In: 2012 ICPR international workshop on pattern recognition and crowd analysis

Angel Arul Jothi J, Mary Anita Rajam V (2017) A survey on automated cancer diagnosis from histopathology images. Artif Intell Rev 48(1):31–81. doi:10.1007/s10462-016-9494-6

Aubry M, Russell BC (2015) Understanding deep features with computer-generated imagery. In: IEEE international conference on computer vision, pp 2875–2883. doi:10.1109/ICCV.2015.329

Bainbridge WS (2007) The scientific research potential of virtual worlds. Science 317(5837):472–476. doi:10.1126/science.1146930

Bertozzi M, Broggi A (1998) GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Trans Image Process 7(1):62–81. doi:10.1109/83.650851

Brutzer S, Höferlin B, Heidemann G (2011) Evaluation of background subtraction techniques for video surveillance. IEEE conference on computer vision and pattern recognition, pp 1937–1944. doi:10.1109/CVPR.2011.5995508

Butler DJ, Wulff J, Stanley GB, Black MJ (2012) A naturalistic open source movie for optical flow evaluation. In: 2012 European conference on computer vision, pp 611–625. doi:10.1007/978-3-642-33783-3_44

Caltech Pedestrian Detection Benchmark. http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/. Accessed 26 June 2017

Cappelli R (2015) Fingerprint sample synthesis. In: Li SZ, Jain AK (eds) Encyclopedia of biometrics, 2nd ed. Springer, New York, pp 668–679

Charalambous CC, Bharath AA (2016) A data augmentation methodology for training machine/deep learning gait recognition algorithms. In: 2016 British Machine Vision conference. doi:10.5244/C.30.110

Chen C, Seff A, Kornhauser A, Xiao J (2015) DeepDriving: learning affordance for direct perception in autonomous driving. In: 2015 IEEE international conference on computer vision, pp 2722–2730. doi:10.1109/ICCV.2015.312

Chen W, Wang H, Li Y, Su H, Wang Z, Tu C, Lischinski D, Cohen-Or D, Chen B (2016) Synthesizing training images for boosting human 3D pose estimation. arXiv:1604.02703

Cheung E, Wong TK, Beral A, Wang X, Manocha D (2016) LCrowdV: generating labeled videos for simulation-based crowd behavior learning. arXiv:1606.08998

COCO—Common Objects in Context. http://mscoco.org/. Accessed 26 June 2017

Correa M, Ruiz-del-Solar J, Verschae R (2016) A realistic virtual environment for evaluating face analysis systems under dynamic conditions. Pattern Recognit 52:160–173. doi:10.1016/j.patcog.2015.11.008

Courty N, Allain P, Creusot C, Corpetti T (2014) Using the AGORASET dataset: assessing for the quality of crowd video analysis methods. Pattern Recognit Lett 44:161–170. doi:10.1016/j.patrec.2014.01.004

Creusot C, Courty N (2013) Ground truth for pedestrian analysis and application to camera calibration. In: IEEE conference on computer vision and pattern recognition workshops, pp 712–718. doi:10.1109/CVPRW.2013.108

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE conference on computer vision and pattern recognition, pp 886–893. doi:10.1109/CVPR.2005.177
Danielsson O, Aghazadeh O (2014) Human pose estimation from RGB input using synthetic training data. arXiv:1405.1213
Datondji SRE, Dupuis Y, Subirats P, Vasseur P (2016) A survey of vision-based traffic monitoring of road intersections. IEEE Trans Intell Transp Syst 17(10):2681–2698. doi:10.1109/TITS.2016.2530146
Dosovitskiy A, Springenberg JT, Tatarchenko M, Brox T (2017) Learning to generate chairs, tables and cars with convolutional networks. IEEE Trans Pattern Anal Mach Intell 39(4):692–705. doi:10.1109/TPAMI.2016.2567384
Farabet C, Couprie C, Najman L, LeCun Y (2013) Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell 35(8):1915–1929. doi:10.1109/TPAMI.2012.231
Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645. doi:10.1109/TPAMI.2009.167
Fernández C, Baiget P, Roca FX, Gonzàlez J (2011) Augmenting video surveillance footage with virtual agents for incremental event evaluation. Pattern Recognit Lett 32:878–889. doi:10.1016/j.patrec.2010.09.027
Ferrer MA, Diaz-Cabrera M, Morales A (2015) Static signature synthesis: a neuromotor inspired approach for biometrics. IEEE Trans Pattern Anal Mach Intell 37(3):667–680. doi:10.1109/TPAMI.2014.2343981
Gaidon A, Wang Q, Cabon Y, Vig E (2016) Virtual worlds as proxy for multi-object tracking analysis. In: IEEE conference on computer vision and pattern recognition, pp 4340–4349. doi:10.1109/CVPR.2016.470
Galbally J, Plamondon R, Fierrez J, Ortega-Garcia J (2012a) Synthetic on-line signature generation. Part I: methodology and algorithms. Pattern Recognit 45:2610–2621. doi:10.1016/j.patcog.2011.12.011
Galbally J, Fierrez J, Ortega-Garcia J, Plamondon R (2012b) Synthetic on-line signature generation. Part II: experimental validation. Pattern Recognit 45:2622–2632. doi:10.1016/j.patcog.2011.12.007
Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(59):1–35
Ghifary M (2016) Domain adaptation and domain generalization with representation learning. Dissertation, Victoria University of Wellington, New Zealand
Gopalan R, Li R, Patel VM, Chellappa R (2015) Domain adaptation for visual recognition. Found Trends® in Comput Graph Vis 8(4):285–378. doi:10.1561/0600000057
Gou C, Wang K, Yao Y, Li Z (2016) Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines. IEEE Trans Intell Transp Syst 17(4):1096–1107. doi:10.1109/TITS.2015.2496545
Gould S, Rodgers J, Cohen D, Elidan G, Koller D (2008) Multi-class segmentation with relative location prior. Int J Comput Vision 80(3):300–316. doi:10.1007/s11263-008-0140-x
Goyette N, Jodoin PM, Porikli F, Konrad J, Ishwar P (2014) A novel video dataset for change detection benchmarking. IEEE Trans Image Process 23(11):4663–4679. doi:10.1109/TIP.2014.2346013
Gupta A, Vedaldi A, Zisserman A (2016) Synthetic data for text localisation in natural images. In: 2016 IEEE conference on computer vision and pattern recognition, pp 2315–2324. doi:10.1109/CVPR.2016.254
Halim Z, Kalsoom R, Bashir S, Abbas G (2016) Artificial intelligence techniques for driving safety and vehicle crash prediction. Artif Intell Rev 46(3):351–387. doi:10.1007/s10462-016-9467-9
Haltakov V, Unger C, Ilic S (2013) Framework for generation of synthetic ground truth data for driver assistance applications. In: 35th German conference on pattern recognition. doi:10.1007/978-3-642-40602-7_35
Handa A, Pătrăucean V, Badrinarayanan V, Stent S, Cipolla R (2015) SceneNet: understanding real world indoor scenes with synthetic data. arXiv:1511.07041
Handa A, Pătrăucean V, Badrinarayanan V, Stent S, Cipolla R (2016) Understanding real world indoor scenes with synthetic data. In: IEEE conference on computer vision and pattern recognition, pp 4077-4085. doi:10.1109/CVPR.2016.442
Hattori H, Boddeti VN, Kitani K, Kanade T (2015) Learning scene-specific pedestrian detectors without real data. In: IEEE conference on computer vision and pattern recognition, pp 3819–3827. doi:10.1109/CVPR.2015.7299006
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, pp 770–778. doi:10.1109/CVPR.2016.90
ImageNet. http://www.image-net.org/. Accessed 26 June 2017
INRIA Person Dataset. http://pascal.inrialpes.fr/data/human/. Accessed 26 June 2017
Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2016) Reading text in the wild with convolutional neural networks. Int J Comput Vis 116:1–20. doi:10.1007/s11263-015-0823-z
Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2014) Synthetic data and artificial neural networks for natural scene text recognition. arXiv:1406.2227

Johnson-Roberson M, Barto C, Mehta R, Sridhar SN, Vasudevan R (2016) Driving in the matrix: can virtual worlds replace human-generated annotations for real world tasks? arXiv:1610.01983
Jones N (2014) Computer science: the learning machines. Nature 505(7482):146–148
Kaneva B, Torralba A, Freeman WT (2011) Evaluation of image features using a photorealistic virtual world. In: 2011 IEEE international conference on computer vision, pp 2282–2289. doi:10.1109/ICCV.2011.6126508
Karamouzas I, Overmars M (2012) Simulating and evaluating the local behavior of small pedestrian groups. IEEE Trans Vis Comput Gr 18(3):394–406. doi:10.1109/TVCG.2011.133
Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, vol 25 (NIPS 2012). doi:10.1145/3065386
LeCun Y, Bengio Y, Hinton GE (2015) Deep learning. Nature 521(7553):436–444. doi:10.1038/nature14539
Liu Y, Wang K, Shen D (2016) Visual tracking based on dynamic coupled conditional random field model. IEEE Trans Intell Transp Syst 17(3):822–833. doi:10.1109/TITS.2015.2488287
Loshchilov I, Hutter F (2016) Online batch selection for faster training of neural networks. arXiv:1511.06343
Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, Corke P, Milford MJ (2016) Visual place recognition: a survey. IEEE Trans Rob 32(1):1–19. doi:10.1109/TRO.2015.2496823
Luo J, Tang J, Tjahjadi T, Xiao X (2016) Robust arbitrary view gait recognition based on parametric 3D human body reconstruction and virtual posture synthesis. Pattern Recognit 60:361–377. doi:10.1016/j.patcog.2016.05.030
Mahendran A, Bilen H, Henriques JF, Vedaldi A (2016) ResearchDoom and CocoDoom: learning computer vision with games. arXiv:1610.02431
Marín J, Vázquez D, Gerónimo D, López AM (2010) Learning appearance in virtual scenarios for pedestrian detection. In: 2010 IEEE conference on computer vision and pattern recognition, pp 137–144. doi:10.1109/CVPR.2010.5540218
Mayer N, Ilg E, Häusser P, Fischer P, Cremers D, Dosovitskiy A, Brox T (2016) A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE conference on computer vision and pattern recognition, pp 4040–4048. doi:10.1109/CVPR.2016.438
Model I, Shamir L (2015) Comparison of data set bias in object recognition benchmarks. IEEE Access 3:1953–1962. doi:10.1109/ACCESS.2015.2491921
Movshovitz-Attias Y, Kanade T, Sheikh Y (2016) How useful is photo-realistic rendering for visual learning? arXiv:1603.08152
Neves J, Narducci F, Barra S, Proença H (2016) Biometric recognition in surveillance scenarios: a survey. Artif Intell Rev 46(4):515–541. doi:10.1007/s10462-016-9474-x
Peng X, Sun B, Ali K, Saenko K (2015) Learning deep object detectors from 3D models. In: 2015 IEEE international conference on computer vision, pp 1278–1286. doi:10.1109/ICCV.2015.151
Pepik B, Benenson R, Ritschel T, Schiele B (2015) What is holding back convnets for detection? arXiv:1508.02844
Pinto N, Barhomi Y, Cox DD, DiCarlo JJ (2011) Comparing state-of-the-art visual features on invariant object recognition tasks. In: IEEE workshop on applications of computer vision, pp 463–470. doi:10.1109/WACV.2011.5711540
Prendinger H, Gajananan K, Zaki AB, Fares A, Molenaar R, Urbano D, van Lint H, Gomaa W (2013) Tokyo Virtual Living Lab: designing smart cities based on the 3D Internet. IEEE Internet Comput 17(6):30–38. doi:10.1109/MIC.2013.87
Qiu W, Yuille A (2016) UnrealCV: connecting computer vision to Unreal Engine. In: 2016 ECCV workshop on virtual/augmented reality for visual artificial intelligence, pp 909-916. doi:10.1007/978-3-319-49409-8_75
Qureshi F, Terzopoulos D (2008) Smart camera networks in virtual reality. Proc IEEE 96(10):1640–1656. doi:10.1109/JPROC.2008.928932
Ragheb H, Velastin S, Remagnino P, Ellis T (2008) ViHASi: virtual human action silhouette data for the performance evaluation of silhouette-based action recognition methods. In: ACM/IEEE international conference on distributed smart cameras, pp 1–10. doi:10.1109/ICDSC.2008.4635730
Ramezani M, Yaghmaee F (2016) A review on human action analysis in videos for retrieval applications. Artif Intell Rev 46(4):485–514. doi:10.1007/s10462-016-9473-y
Rematas K, Ritschel T, Fritz M, Tuytelaars T (2014) Image-based synthesis and re-synthesis of viewpoints guided by 3D models. In: 2014 IEEE conference on computer vision and pattern recognition, pp 3898–3905. doi:10.1109/CVPR.2014.498
Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. doi:10.1109/TPAMI.2016.2577031
Ren X, Chen K, Sun J (2016) A CNN based scene Chinese text recognition algorithm with synthetic data engine. arXiv:1604.01891

Richter SR, Vineet V, Roth S, Koltun V (2016) Playing for data: ground truth from computer games. In: 2016 European conference on computer vision, pp 102–118. doi:10.1007/978-3-319-46475-6_7
Ros G, Sellart L, Materzynska J, Vazquez D, Lopez AM (2016) The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: 2016 IEEE conference on computer vision and pattern recognition, pp 3234–3243. doi:10.1109/CVPR.2016.352
Rozantsev A, Lepetit V, Fua P (2015) On rendering synthetic images for training an object detector. Comput Vis Image Underst 137:24–37. doi:10.1016/j.cviu.2014.12.006
Shafaei A, Little JJ, Schmidt M (2016) Play and learn: using video games to train computer vision models. In: 2016 The British machine vision conference. doi:10.5244/C.30.26
Shotton J, Girshick R, Fitzgibbon A, Sharp T, Cook M, Finocchio M, Moore R, Kohli P, Criminisi A, Kipman A, Blake A (2013) Efficient human pose estimation from single depth images. IEEE Trans Pattern Anal Mach Intell 35(12):2821–2840. doi:10.1109/TPAMI.2012.241
Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. arXiv:1604.03540
Sizikova E, Singh VK, Georgescu B, Halber M, Ma K, Chen T (2016) Enhancing place recognition using joint intensity—depth analysis and synthetic data. In: ECCV workshop on virtual/augmented reality for visual artificial intelligence, pp 901–908. doi:10.1007/978-3-319-49409-8_74
Smelik RM, Tutenel T, Bidarra R, Benes B (2014) A survey on procedural modeling for virtual worlds. Comput Graphics Forum 33(6):31–50. doi:10.1111/cgf.12276
Sobral A, Vacavant A (2014) A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput Vis Image Underst 122:4–21. doi:10.1016/j.cviu.2013.12.005
Starzyk W, Qureshi F (2013) Software laboratory for camera networks research. IEEE J Emerg Select Top Circuits Syst 3(2):284–293. doi:10.1109/JETCAS.2013.2256827
Sun B, Peng X, Saenko K (2015) Generating large scale image datasets from 3D CAD models. In: CVPR 2015 Workshop on the future of datasets in vision
Sun B, Saenko K (2014) From virtual to reality: fast adaptation of virtual object detectors to real domains. In: 2014 British machine vision conference. doi:10.5244/C.28.82
Su H, Qi CR, Li Y, Guibas L (2015) Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views. In: IEEE international conference on computer vision, pp 2686–2694. doi:10.1109/ICCV.2015.308
Szeliski R (2010) Computer vision: algorithms and applications. Springer, New York
Taylor GR, Chosak AJ, Brewer PC (2007) OVVV: using virtual worlds to design and evaluate surveillance systems. In: 2007 IEEE conference on computer vision and pattern recognition, pp 1–8. doi:10.1109/CVPR.2007.383518
Thacker NA, Clark AF, Barron JL, Beveridge JR, Courtney P, Crum WR, Ramesh V, Clark C (2008) Performance characterization in computer vision: a guide to best practices. Comput Vis Image Underst 109(3):305–334. doi:10.1016/j.cviu.2007.04.006
The KITTI Vision Benchmark Suite. http://www.cvlibs.net/datasets/kitti/. Accessed 26 June 2017
The PASCAL Visual Object Classes homepage. http://host.robots.ox.ac.uk/pascal/VOC/. Accessed 26 June 2017
Torralba A, Efros AA (2011) Unbiased look at dataset bias. In: 2011 IEEE conference on computer vision and pattern recognition, pp 1521–1528. doi:10.1109/CVPR.2011.5995347
Vacavant A, Chateau T, Wilhelm A, Lequièvre L (2013) A benchmark dataset for outdoor foreground background extraction. In: ACCV 2012 workshops, Lecture Notes in Computer Science vol 7728, pp 291–300. doi:10.1007/978-3-642-37410-4_25
Vázquez D (2013) Domain adaptation of virtual and real worlds for pedestrian detection. Dissertation, Universitat de Barcelona, Spain
Vázquez D, López AM, Marín J, Ponsa D, Gerónimo D (2014) Virtual and real world adaptation for pedestrian detection. IEEE Trans Pattern Anal Mach Intell 36(4):797–809. doi:10.1109/TPAMI.2013.163
Veeravasarapu VSR, Hota RN, Rothkopf C, Visvanathan R (2015a) Model validation for vision systems via graphics simulation. arXiv:1512.01401
Veeravasarapu VSR, Hota RN, Rothkopf C, Visvanathan R (2015b) Simulations for validation of vision systems. arXiv:1512.01030
Veeravasarapu VSR, Rothkopf C, Visvanathan R (2016) Model-driven simulations for deep convolutional neural networks. arXiv:1605.09582
Venetianer PL, Deng H (2010) Performance evaluation of an intelligent video surveillance system - a case study. Comput Vis Image Underst 114(11):1292–1302. doi:10.1016/j.cviu.2010.07.010
Wang F-Y (2004) Parallel system methods for management and control of complex systems. Control Decis 19(5):485–489

Wang F-Y (2010) Parallel control and management for intelligent transportation systems: concepts, architectures, and applications. IEEE Trans Intell Transp Syst 11(3):630–638. doi:10.1109/TITS.2010.2060218
Wang F-Y (2013) Parallel control: a method for data-driven and computational control. Acta Automatica Sinica 39(4):293–302. doi:10.3724/SP.J.1004.2013.00293
Wang K, Huang W, Tian B, Wen D (2012) Measuring driving behaviors from live video. IEEE Intell Syst 27(5):75–80. doi:10.1109/MIS.2012.100
Wang X, Wang M, Li W (2014) Scene-specific pedestrian detection for static video surveillance. IEEE Trans Pattern Anal Mach Intell 36(2):361–374. doi:10.1109/TPAMI.2013.124
Wang F-Y, Wang X, Li L, Li L (2016) Steps toward parallel intelligence. IEEE/CAA J Automatica Sinica 3(4):345–348. doi:10.1109/JAS.2016.7510067
Wang K, Liu Y, Gou C, Wang F-Y (2016) A multi-view learning approach to foreground detection for traffic surveillance applications. IEEE Trans Veh Technol 65(6):4144–4158. doi:10.1109/TVT.2015.2509465
Wang F-Y, Zhang JJ, Zheng X, Wang X, Yuan Y, Dai X, Zhang J, Yang L (2016) Where does AlphaGo go: from Church-Turing Thesis to AlphaGo Thesis and beyond. IEEE/CAA J Automatica Sinica 3(2):113–120. doi:10.1109/JAS.2016.7471613
Wang F-Y, Zhang J, Wei Q, Zheng X, Li L (2017) PDP: parallel dynamic programming. IEEE/CAA J Automatica Sinica 4(1):1–5. doi:10.1109/JAS.2017.7510310
Wang K, Yao Y (2015) Video-based vehicle detection approach with data-driven adaptive neuro-fuzzy networks. Int J Pattern Recognit Artif Intell. doi:10.1142/S0218001415550150
Wulff J, Butler DJ, Stanley GB, Black MJ (2012) Lessons and insights from creating a synthetic optical flow benchmark. In: 2012 ECCV workshop on unsolved problems in optical flow and stereo estimation, pp 168–177. doi:10.1007/978-3-642-33868-7_17
Xu J, Vázquez D, López AM, Marín J, Ponsa D (2014) Learning a part-based pedestrian detector in a virtual world. IEEE Trans Intell Transp Syst 15(5):2121–2131. doi:10.1109/TITS.2014.2310138
Xu J, Ramos S, Vázquez D, López AM (2014) Domain adaptation of deformable part-based models. IEEE Trans Pattern Anal Mach Intell 36(12):2367–2380. doi:10.1109/TPAMI.2014.2327973
Yang L, Wang F-Y (2007) Driving into intelligent spaces with pervasive communications. IEEE Intell Syst 22(1):12–15. doi:10.1109/MIS.2007.8
Zeng X, Ouyang W, Wang M, Wang X (2014) Deep learning of scene-specific classifier for pedestrian detection. In: 2014 European conference on computer vision, pp 472-487. doi:10.1007/978-3-319-10578-9_31
Zhang N, Wang F-Y, Zhu F, Zhao D, Tang S (2008) DynaCAS: computational experiments and decision support for ITS. IEEE Intell Syst 23(6):19–23. doi:10.1109/MIS.2008.101
Zhu W, Wang F-Y (2012) The fourth type of covering-based rough sets. Inf Sci 201:80–92. doi:10.1016/j.ins.2012.01.026
Zitnick CL, Vedantam R, Parikh D (2016) Adopting abstract images for semantic scene understanding. IEEE Trans Pattern Anal Mach Intell 38(4):627–638. doi:10.1109/TPAMI.2014.2366143
Zuo J, Schmid NA, Chen X (2007) On generation and analysis of synthetic iris images. IEEE Trans Inf Forensics Secur 2(1):77–90. doi:10.1109/TIFS.2006.890305