Synthetic data and simulation

for real-world business advantage

Contents

Foreword 02

1 Data collection and training: an automotive example 03

2 Managing difficult interactions – and development costs 05

Case study 1: domestic automation 05

3 The edge of reason: reducing risk in dangerous environments 08

4 Accelerating R&D to get to market quicker 09

Case study 2: autonomous vehicle example 11

Next steps 12

Authors 12

Foreword

Simulation techniques were already changing the world. But as society adjusts to the effects of the pandemic, a fresh and forceful spirit of invention is gaining momentum. Massive transformation is afoot in product evolution and R&D, much of it driven by the promise of AI. But as we place greater trust in algorithms to help decision making and processes we need to invest carefully. The wrong turns could be expensive and time consuming. This whitepaper explores how the right combination of low-cost simulation environments and machine learning techniques can massively reduce time to market, without breaking the bank.

Our essential proposition is simple. Ambitious companies that invest in synthetic data and simulation environments will be rewarded with product research and development that is drastically cheaper and faster.

They will jump ahead of the competition and reap the rewards. This applies to all projects but with particular relevance to those with elements of control, automation and intuitive interactions. Projects of this nature deploy a rich vein of machine intelligence from across the broad spectrum of artificial intelligence (AI), specifically machine learning (ML) and deep learning (DL) techniques. Typically success depends on feeding in huge volumes of real-world data (RWD). The task of creating or generating the RWD is extremely expensive, time consuming and in some cases impossible – it may reside in a dangerous environment or simply not exist in the world.

The solution is to create an environment that is true in nature to the one in which the system is going to perform. Such synthetic data environments allow businesses to reduce development costs, risk and time to market, as well as enabling the seemingly impossible. Uncharted areas of application are opened up. One robot might enter and clean a hazardous environment with limited information about what's inside. Another might navigate snowy streets and pavements to deliver a parcel.

In both cases, it would be prohibitively slow and expensive to test and train the robot in real life. But complementing a degree of in-the-field training with synthetic data allows an innovator to test their solution in thousands-upon-thousands of simulations. It speeds their development process and gets them to market before their competitors. But this isn't all about robots.

The once-traditional automotive industry has already begun to explore the use of simulated environments and mature gaming engine platforms. These technologies are needed to develop algorithms for scene understanding. Both have been invaluable in building level 2 autonomous driving (partial, with the driver engaged with the driving task) and are essential for level 3 (the driver is not required to monitor the environment).

This discipline of developing and training certain types of algorithm has been maturing over the last decade. A machine learning adage has it that life begins at one hundred thousand samples. And as increasingly sophisticated applications emerged – with ever more complex edge cases – it was obviously unrealistic to collect such vast amounts of real-world-data. Hence the importance of simulation and synthetic data, which now takes its place as a vital asset for businesses responding to the challenges imposed by the pandemic.

As we write, the economic effects of the pandemic are only starting; worse is to come. But regulatory deadlines, such as EU emission targets, remain in place. Consumer purchasing behaviour will never be the same again. Yet investors expect and demand rapid gains and will dump the stock of organisations that are inflexible to change. The C-suite will apply increasing pressure on their R&D and manufacturing capabilities. And why wouldn't they? The Covid crisis proved that multi-year development cycles for complex products such as ventilators could be shortened to a matter of months.

We will explain how the use of simulation environments and synthetic data – and similar technologies that are aiding in training reinforcement learning agents – opens the door to a myriad of solutions across industries more quickly and cost effectively than ever before. This simulation is real; time to get onboard.

Niall Mottram
Head of Industrial & Energy
niall.mottram@cambridgeconsultants.com

1 Data collection and training: an automotive example

If the ambition is to get to market first with reduced cost and risk, then the need for speed is linked to the need for accurate, real-world data. The amount of data needed to train a machine learning model varies by domain and by model. In most cases the numbers get big very quickly. This can become expensive and a barrier to entry for new industry players and start-ups. For non-trivial problems, the expectation is that hundreds of thousands of samples will be needed. The more subtle the problem, the more data is needed for training.

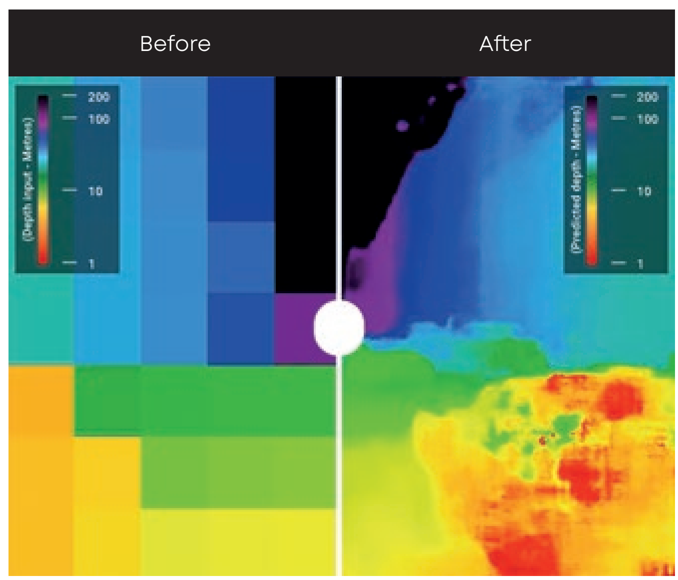
If generating data is a manual process, collecting such a volume of data becomes expensive. We recently encountered such a challenge when developing our simulation tools for the testing of navigation algorithms and ADAS sensors for both automotive and off-highway environments. We were developing an algorithm to combine low quality depth data with high quality RGB input.

The combination of these sensors allowed us to produce high-detail depth data at a far lower cost by avoiding the need to use a costly depth sensor. In order to train the model, we needed depth and RGB images in multiple environments and lighting conditions. To collect this data from real sensors would have been prohibitively expensive. If there were no alternative, we would have been unable to show the validity of our neural network design. But we were able to generate endless data by creating a 3D virtual environment. We found 60,000 images was the minimum to obtain good results.

The number of images necessary to train a neural network is related to the complexity of the model. A simple input will have many similarities between samples. The more variables in the input data, the more complex a model is needed and therefore the more images will be needed to adequately train the neural network. Unless the training data matches the real data that the algorithm is expected to encounter in use, the network is likely to be overfitted to the training set – which means that great performance against the training data will not be realised in a real-world situation.

It will not generalise well. The number of training images necessary is related to the complexity of the problem domain on which the network is expected to perform. To think about this a different way, the ability to generate more and better training data enables us to create algorithms which are applicable to a wider variety of problems.

Adding one or two more variables to the capability of an algorithm can make its operation much more effective. In some domains the increase in reliability can lead to greater yields or even save lives. It also causes an explosion in the amount of necessary training resources. This underlines the critical importance of developing relevant synthetic data.



AI model achieves a significant uplift in depth resolution using novel algorithms to fuse low-cost sensors

"The ability to generate more and better training data enables us to create algorithms which are applicable to a wider variety of problems."

Synthetic data and virtual learning environments bring further advantages. Often, labeling the data from real world cameras and sensors is more work and expense than capturing the data in the first place, and these labels may themselves be incorrect. In contrast, synthetic data can be perfectly labelled, and with a precision which is otherwise impossible. If required, to more accurately model the real world, sources of noise can be applied in a controlled way.

Another significant advantage to investing in simulation as a production environment for training reinforcement learning algorithms is that the same simulated environment can be reused on multiple algorithms. Changes to requirements do not require a complete reinvestment of resources and time to generate a new set of data. The business advantages provided by these changes to cost structures are explored later in the paper.

Virtual environments reduce development risk and time to market, while increasing the size of the addressable market. But they require the ability to accurately model the world, understand the sensor and construct the environment. Doing this negates the pain of physically building huge numbers of objects and environments for your system to handle. You are free to test the range of objects and environments in simulation at many times the speed, and then restrict the more expensive real-world testing to confirming the fidelity of your simulation. You can also speed your time to market by reducing development cycles if you can precisely repeat your test.

For example, if your system fails just when sunlight hits a sensor at a specific angle, that might be impossible or at least extremely time consuming to recreate in the real world. But it's simple to replay any exact scenario over and over in a digital environment until you find the problem. Finally, by broadening the scenarios you can test during development, you can get a product that is known to work well in many markets.

"How do product engineers build confidence that the real-world system will behave in the same way as the simulation?"

2 Managing difficult interactions – and development costs

Labour shortages have been impacting several industries for some time. Agriculture was increasingly struggling to find labourers for harvesting activities, and the hospitality industry saw fewer and fewer applicants for housekeeping, cooking and catering roles. With the need for increased social distancing added to the mix, there is a greater drive than ever to augment the labour force with intelligent automation and robotics.

Such robotic systems must be reactive to their environments in a way that for humans is second nature. Much like a toddler learns by repeating processes over and over again, so AI can bring new intelligence to previously dumb automation systems. An agent which learns by repeated interaction with an environment, receiving feedback about its success, is called a reinforcement learning agent. Reinforcement learning agents show great promise for solving some valuable problems, however, the commercial justification for the costs of the real-world data collection has been challenging. This is made worse by needing to iterate the hardware and environmental parameters and subsequently regather new data.

Methods are emerging from academia for running virtual simulations to gather the volume of data required for training reinforcement learning models. Using these techniques successfully requires overcoming complex challenges around real-world/virtual-world equivalency. How do product engineers build confidence that the real-world system will behave in the same way as the simulation? We set about investigating and answering this question by creating a virtually trained system with a physical hardware twin. The example use case was clearing a wide variety of items from a living room floor using a robotic arm and gripper.

Case study 1: domestic automation

We built the virtual half of the twin using 3D data files for the individual components of the robotic arm and gripper, recreating them in the simulated scene to full scale. The full kinematics used by the real robotic arm to determine how the joints need to be rotated to put the gripper in a given pose were calculated, ensuring the simulated robot replicated the real robot's movement. The algorithm was implemented directly in C# to run within the simulation engine. The goal was to be unable to distinguish between virtual and real twins.

Enabling the accuracy and completeness of the physics in the simulation was very important in reaching this goal. We were dealing with moving physical objects and needed the simulation to behave as much as possible like reality. Friction in particular was a vital behaviour to model correctly due to the types of objects to be handled and the gripper design. A game engine was selected that uses Nvidia's powerful PhysX physics engine. It is able to accurately simulate all the relevant physical effects, including friction. Such a game engine prioritises a high framerate, meaning it can simulate the physics and the camera data very quickly. Simulations can be run at up to 100x real time, allowing faster data generation, model training and verification. Furthermore, many simulations can be run in parallel, delivering the final output in elapsed times that could never be practically realised in real-world testing.

Simulation of the objects to be picked did not require the same fidelity compared to the simulated robot. The goal here was to train a network that is as general as possible, meaning it should pick up objects it has never seen before. The objects therefore only needed to be reasonably representative. They should be similar sizes, shapes, topologies and so on, but need not be exact replicas of real objects.

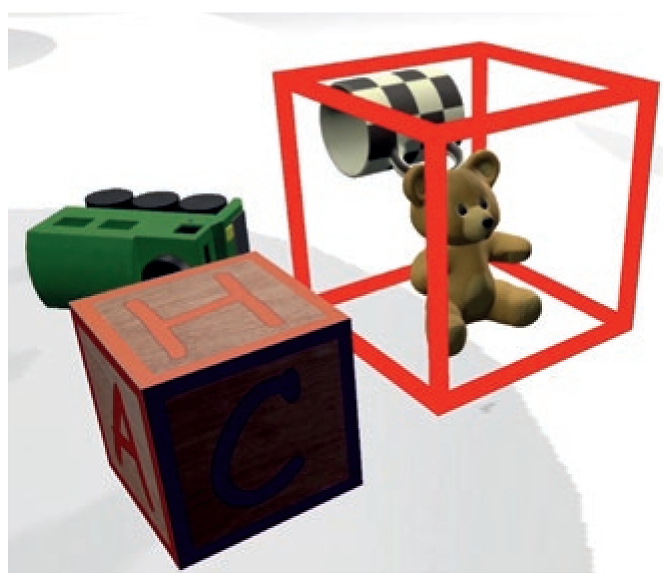


Figure 1: Items for virtual training rendered in the digital twin

The simulated virtual environment, hardware, sensors and objects running on the game engine powered multiple aspects of the final solution:

1. The collection of human user demonstration data (in VR within the environment)
2. The generation of substantially more training data
3. The validation of the trained model on the simulated robot

Leveraging the developed system across multiple facets of the task builds its value, making a clearer case for investment.

Human demonstrations were collected as a starting point for the grip prediction model training data set. This use of human demonstrations in training a model to complete a task is referred to as imitation learning. It is useful because it gives the model a more advanced starting point, leapfrogging the need to simulate all possible approaches at the outset. The required output data was twofold, both a position and orientation at which the gripper could pick up a given object successfully. A mixture of human users picked up a set of objects within the simulation, using a VR headset and controllers to orient and position the simulated gripper.

Only a small number of these human demonstrations (tens) were required, as the game engine could then generate many more picking attempts. Each generated picking attempt randomised the position and orientation of each item in the scene, allowed simulated gravity to settle the object in a realistic pose, then applied the human demonstration grips for the object. These were checked for feasibility via a set of collision checks which in further work would be extended to include simulating the full gripping action with the virtual robot arm and assessing the success of the grip. Data for all picking attempts was saved (including the success or failure label) and used in training the model.

After training the model to predict the best gripper pose to pick up an object seen in the depth cameras, the simulation can quickly validate its performance without having to conduct real-world tests. As with the data generation, the validation can be run many times faster than the real-world system. Furthermore, the validation scenes are exactly replicable, meaning the results are directly comparable to one another, rarely the case in real lab testing. It is also much easier to record exactly what is happening and what went wrong when the model fails in the virtual validation. Error modes like self-collision, collision with other objects, dropping and so on are easily detected and completely repeatable, meaning improvements are much easier to track and verify.

Training a model that receives an occupancy grid from 3D depth cameras and outputs a successful grip pose is a complex problem that requires careful algorithm design. A black-box 3D CNN (Convolution Neural Network) trained with a generic loss function such as MSE (Mean Squared Error) typically fails to produce useful regression estimates for several reasons, including:

1. Limitations on the quantity of manual grips that can realistically be collected on many different types of complex object, even in a simulated environment
2. The indeterminacy that arises from objects' rotational and reflective symmetries

Without a large amount of data on many different 3D forms, a poorly designed network with moderate capacity tends simply to memorise the shapes it has seen in training and shows poor generalisation performance. Avoiding this problem requires an efficient feature extraction component that reduces a complex shape to the bare minimum required to estimate a grip.

We further observed that loss functions that are unaware of symmetries are liable to generate strong error signals on valid grip configurations unless all valid configurations have been explicitly labelled – a tedious task. In experiments, such loss terms not only rendered the network unable to learn, but actually biased it away from the grip orientations naturally selected by humans.

To address these issues, we designed a feature extraction model based on Mixture Density Networks and created a custom loss function in Tensorflow. This allowed additional parameters such as an object's symmetries to be passed into the function alongside its usual parameters, enabling the loss calculation to adapt based on geometric information.

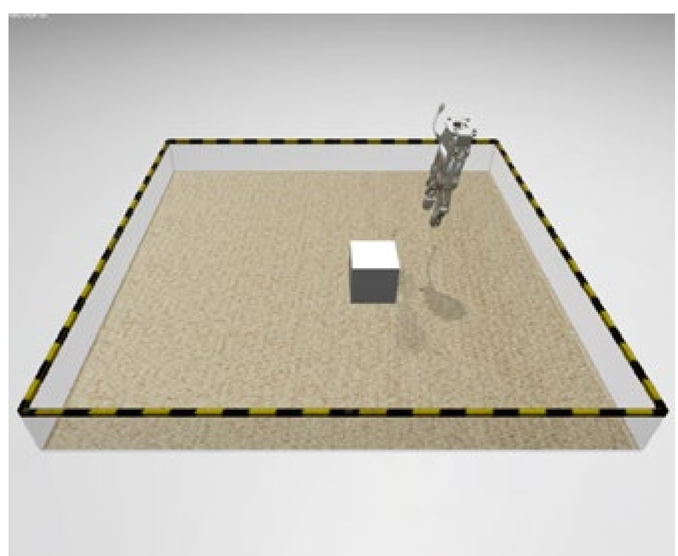


Figure 2: Training environment with physics engine and VR to collect "human demonstrations"

Pre-trained backbone networks that can be exploited for transfer learning are much less easily available for 3D CNNs than for their 2D counterparts. In the absence of a reliable pre-trained network, we initialised the weights of the feature extraction model by training it to learn just the position, size and pose of each object's principal axes.

Using a low-dimensional, physically interpretable intermediate state in this way provides several benefits. It allows performance to be investigated and understood at a much lower level than is possible within a black-box network architecture. It also helps to modularise the development process, making development more understandable, better controlled and closer to the mature paradigms of software engineering rather than the all-or-nothing nature of many machine learning developments.

In this case however, the greatest benefit was that a feature extraction module with such simple interfaces can be pre-trained on even simpler data than that collected in the synthetic environment. Just by training on a set of artificial cuboids in a variety of positions, sizes and orientations the module was able to accurately predict these key geometric attributes of an object. Crucially, we showed that its performance generalised well to more complex shapes.

Having successfully trained an effective 3D feature extractor, we could use it as a foundation to extend the system component by component. We increased the richness of the network's internal representation by adding a 2D feature extractor based on MobileNet to process RGB images of the object and fed both sets of features into a data fusion component that learned to estimate grips.

Then we fine-tuned the network by training end-to-end on the detailed but relatively scarce manual grip data.

We found that an approach making use of abundant simplistic data to pre-train at the level of system components, followed by system-level integration and tuning, allowed us to develop a machine learning system that could accurately identify valid grips on a wide range of objects it had never been exposed to in training.

We can conclude that informed design at the subsystem level will not achieve state-of-the-art performance in situations with unlimited labelled training data, sometimes it may not even be applicable. However, when data is at a premium (which is often the case), our approach described above is clearly an important tool in the AI engineer's workshop. It rapidly builds confidence in the real-virtual world equivalency, helping to lower risk, facilitate oversight and, ultimately, deliver the results required in a commercially feasible manner.

The selected use case for the investigation above – clearing a toy-covered living room floor – replicates a frustrating, but ultimately benign task. However, simulation environments have a significant role to play when the opposite is true. Occupational risk and therefore commercial risk rise when executing developments for systems in dangerous and hazardous environments.

Now we will explore how simulation could play a key role in overcoming these challenges.

"Occupational risk and therefore commercial risk rise when executing developments for systems in dangerous and hazardous environments."

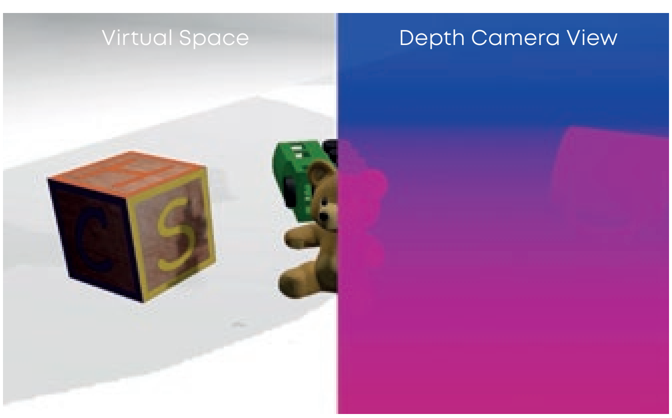


Figure 3: Simulating sensor (depth camera) output in the digital twin

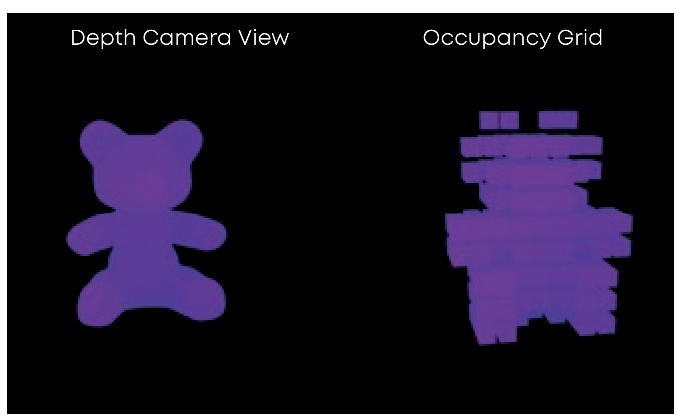


Figure 4: An occupancy grid is calculated from sensor data after segmentation from the scene

3 The edge of reason: reducing risk in dangerous environments

We have established that synthetic data and learning environments can make machine learning techniques feasible for real-world problems. The virtual nature of the data generation makes the technique particularly compelling when dealing with safety concerns. The testing and development that becomes possible should attract wide attention. These edge cases include equipment malfunctions, misuse and freak weather events. For use cases in which failure risks injury or death, such techniques are clearly attractive and are likely to become compulsory.

Let's consider the automation of clean up after tragic incidents such as natural disasters or the destruction of a nuclear plant. Systems must be designed to both reduce the risk of these catastrophes and also to limit the damage as much as possible in the aftermath. Emergency scenarios are difficult to recreate. They would be both prohibitively expensive and dangerous to replicate realistically. Not only are humans in extreme circumstances likely to behave irrationally, but artificial intelligences are also liable to react poorly if these odd circumstances have not been present in their training.

It is of vital importance then, that we consider using simulations of disasters in both learning environments for reinforcement learning agents and the production of synthetic data. Another instance where it is impossible to collect the data for training neural networks is where the environment does not yet exist, is unknown, or will be constantly changing. For example, a mine in which the operation is literally changing the environment.

A simulated environment can ensure that intelligent agents are able to operate sensibly and most efficiently throughout the time the mine is in operation. In fact, if the development of the mine is known in advance, the progression stages can be included in the input data. This will allow algorithms to learn how to cooperate most efficiently at any stage. The optimal behaviour for an individual agent may change as the mine grows. This opens the door to exciting new optimisations which are beyond what would be possible without synthetic data techniques.

These environments present unique challenges to product designers. They will come across objects they've never seen before in totally new environments. But regardless of that, the product still needs to be safe when it bridges from the digital to the physical world.

Simulation of disasters helps intelligent products perform safely

4 Accelerating R&D to get to market quicker

The central thrust of this whitepaper concerns the commercial benefit to business of getting to market faster. In this section, we elaborate on some of the ways virtual environments and synthetic data are able to propel projects and accelerate innovation.

In the race to transform cutting-edge technologies into successful products, the ability to overlap research with product development sets market leaders apart. The limitation of course is cost. Speculative product development work can go to waste if further research reveals a significant limitation of the technology. The benefit of being the first to exploit a new market can be significant but it is not infinite. Techniques which enable a faster and richer line of communication between product development cycles change the calculus. Simulation techniques which accelerate the development and understanding of AI models are an example of this.

The development of a simulated environment can be costly initially. But simulation techniques and assets are often applicable to multiple problems. What's more, the established simulation becomes a powerful tool which enables rapid incremental development. This is transformative. Once a process is iterating quickly, progress becomes measurable and it becomes possible to understand and improve the process in previously unimaginable ways. Barriers to experimentation drop away, allowing the problem space to be explored in a fraction of the time. Pioneers can exploit more of the advantages of their exploration before competitors have the chance to react.

Using simulated environments in algorithmic training and development ensures that a well-developed virtual testing environment can be maintained throughout the various strands of development.

From research and development all the way to the product launch, it is possible to use consistent data. This ensures that the thorough technical understanding won by research investment is not diluted and that, ultimately, end users benefit from a higher quality product. Likewise, the sharing of testing data between hardware, software and design ensures that all teams are on the same page. The consistency of vision is maintained as all input operates a shared operating context.

Perhaps an even stronger incentive to build a bridge from research into development and manufacture is traceability. Establishing safety criteria at the outset of the project, and performing regression tests against these considerations throughout development, means that the safety requirements are as embedded into the work as the functional requirements. Safety critical systems are leading the thinking in this area, as development of Safety of the Intended Functionality (SOTIF) standards in automotive grapple with the new world of regulating advanced software systems.

Consumers have yet to be persuaded to put their lives in the hands of the mysteries of black-box neural networks, but they clearly have the potential to create statistically significant increases in safety. The techniques to make this argument compelling lie in the proper use of simulation during the development process. If these techniques are developed appropriately, regulators and the general public will allow companies to provide these valuable benefits.

Simulations are different in kind from collected data. It should not be ignored that the simulated data allows techniques which are immensely valuable. Because a simulated sensor in combination with an algorithm is attempting to model ground truth, the evaluation of the algorithm is easier when the ground truth is available directly. This is the case with synthetic data. In conjunction with our other research into automotive and automation technologies, we are using simulated environments to evaluate the performance of Simultaneous Location and Mapping (SLAM) algorithms which attempt to understand the shape of the world given only a video stream.

"In the race to transform cutting-edge technologies into successful products, the ability to overlap research with product development sets market leaders apart."

Using a simulated environment to produce the video stream in the previous automotive example allowed us to compare the output of the algorithm to the 3D environment itself. Actual sensor data would have required the additional work to map the environment manually. Some of the deviations between the algorithm and the measured environment would be due to the mapping technique, which would result in poorer training metrics for the algorithm.

Two-dimensional data is the most common type of data to train with. There are of course many other types of data in which an algorithm (reinforcement or otherwise) might be expected to operate. For example, an algorithm operating in a financial context might require a simulated economic environment where the dimensions are not spatial but related to the relative value of commodities. Another example of non-spatial data which we are experienced in is the data used in the prediction of human behaviour.

Case study 2: autonomous vehicle example

Where many components must be integrated into a complex environment, the benefits of synthetic data and virtual learning environments become exponentially stronger. We learned this as we were developing a simulation framework for autonomous vehicles. The operational environment and safety requirements made this a highly challenging project, and we were aiming to develop a framework which was suitable for a wide variety of potential use cases. The system was built of modular decoupled components. For many of these components, a machine learning agent was trained and tested. This was only possible within the budget through the use of a game engine, in our case Unity.

We were able to accommodate a large number of requirements through the well tested and integrated simulation system and did not have to create unique simulations for each of the components of the framework. More significantly, we were able to conduct integration tests using an environment which was not novel to the system at the time of integration. This made the integration process drastically cheaper. Now that the system has been designed in a modular and integrated way, we can swap out a single component and allow our client to test one piece in the context of a fully-developed system. For example, a Tier 1 sensor manufacturer can swap out our sensor model for theirs and learn about the impact of the sensor on vehicle safety using existing vehicle dynamics and perception algorithms.

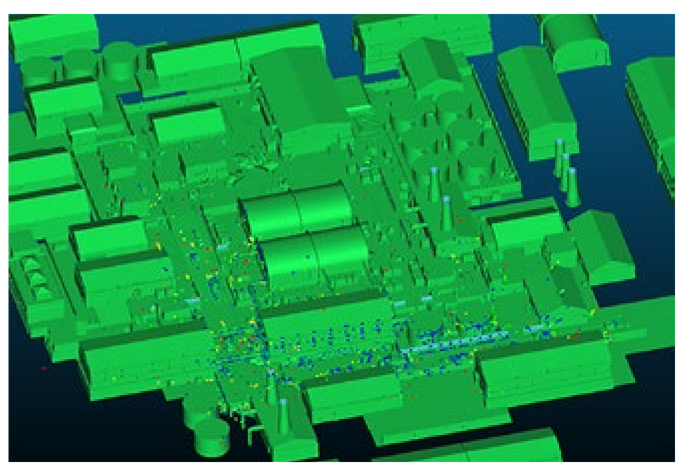
Training reinforcement learning agents with virtual learning environments and neural networks with synthetic data is clearly a powerful technique. But there are limitations. Failure to understand them will result in unexpected delays and increases in spend, potential loss of faith in the approach and even compromises in safety.

Synthetic data approximates real data. Where the synthetic data does not match the real situation, the algorithm being trained may develop incorrect predictions and behaviours. The effectiveness of the algorithm may therefore be limited by the accuracy of the simulation. This is not always the case and although challenging it is usually possible to build simulations around known limitations that ensure that algorithms behave in safe ways. But practitioners must always be seeking to discover the unknown unknowns.

To check that the model does not deviate from reality beyond acceptable bounds, it is useful to consider methods to quantify the effectiveness of the simulation. There are both machine learning and statistical techniques for this problem. For example, techniques used to detect financial fraud are useful to spot irregularities between fake and real data. Generative Adversarial Networks (GANs) are state of the art methods for using AI to generate synthetic data. They comprise a generator and a discriminator. A GAN trained to generate similar data to the simulation will also produce a neural network expert in verifying whether artificial data seems correct. Perhaps adding a machine learning processing step could increase the accuracy of simulated data in the future.

While the project team may be satisfied by the validity of their techniques, it remains necessary to persuade key stakeholders that the use of synthetic data and virtual learning environments is valid. These stakeholders include the key C-suite individuals discussed in the foreword, as well as the customers of the business. Ultimately, where safety is the issue, the concern is risk management. When a safety precaution fails, can a jury be persuaded that the development team were justified in their simulation and training techniques? These are the challenges which will counterbalance the benefits we've discussed when it comes to deciding whether to trust these simulation-based techniques.

The SLAM evaluation framework showing the algorithm's point cloud representation of the scene, corresponding to the location of objects and surfaces observed as the vehicle drives around a simulated industrial scene



Colour coded SLAM points from the algorithm to show how close they align to the true location of objects in the scene from the simulator

Next steps

The trend towards greater automation and autonomy has been gaining momentum for some time. The business case is now even stronger in our new age of social distancing, characterised by the need to augment human intuition and muscle with mechanical intelligence. Simulation environments are a critical part of these developments. They lower development cost, reduce risk and increase speed to market.

At Cambridge Consultants, we continue to work in close partnership with clients, exploring the boundaries of simulation environments across a range of sectors, including:

- **In autonomous vehicles**, providing a competitive edge in the transition through the levels of autonomy
- **In agritech**, searching for solutions in difficult physical environments
- **In marketing and retail**, overcoming GDPR-related consent issues through simulation
- **In clinical and scientific trials**, providing baselines for future studies, where no real data exists
- **In manufacturing**, driving improved performance of quality control systems

These are simply representative examples. In some cases, the use of simulation environments can go even further, levelling the playing field for all in areas where data collection was previously cost prohibitive. As we have seen, in times of crisis humanity can achieve remarkable things. The time has come to be more agile, more flexible, more responsive and – in doing so – become more resilient. Automation will play a part, and simulation environments will be the driver.

Authors

Luke Avery Senior Software Engineer

Chris Roberts Head of Robotics

Henry Fletcher Principal Engineer

About Cambridge Consultants

Cambridge Consultants has an exceptional combination of people, processes, facilities and track record. Brought together, this enables innovative product and services development and insightful technology consulting. We work with companies globally to help them manage the business impact of the changing technology landscape. We're not content to deliver business strategy based on target specifications, published reports or hype. We pride ourselves on creating real value for clients by combining commercial insight with engineering rigor. We work with some of the world's largest blue-chip companies as well as with innovative start-ups that want to change the status quo fast.

With a team of around 800 staff in Cambridge (UK), Boston, San Francisco and Seattle (USA), Singapore and Tokyo, we have all the in-house skills needed to help you – from creating innovative concepts right the way through to taking your product into manufacturing.

For further information or to discuss your requirements, please contact:

Niall Mottram, Head of Industrial & Energy
niall.mottram@cambridgeconsultants.com

cambridge consultants
Part of Capgemini Invent

UK — USA — SINGAPORE — JAPAN

www.cambridgeconsultants.com

Cambridge Consultants is part of Capgemini Invent, the innovation, consulting and transformation brand of the Capgemini Group. www.capgemini.com

Lit-WHI-047 v0.7 '2022_A2A_Synthetic data and simulation for real-world business advantage_Whitepaper'